

JOINT TRANSPORTATION RESEARCH PROGRAM

INDIANA DEPARTMENT OF TRANSPORTATION
AND PURDUE UNIVERSITY



POST-PROCESSING TECHNIQUES TO ENHANCE RELIABILITY OF ASSIGNMENT ALGORITHM BASED PERFORMANCE MEASURES

Srinivas Peeta

Professor of Civil Engineering
Purdue University
Corresponding Author

Amit Kumar

Graduate Research Assistant
School of Civil Engineering
Purdue University

Sushant Sharma

Research Associate
NEXTRANS Center
Purdue University

SPR-3214

Report Number: FHWA/IN/JTRP-2011/19

DOI: 10.5703/1288284314643

This page intentionally left blank.

RECOMMENDED CITATION

Peeta, S., Kumar, A., & Sharma, S. *Post-processing Techniques to Enhance Reliability of Assignment Algorithm Based Performance Measures*. Publication FHWA/IN/JTRP-2011/19. Joint Transportation Research Program, Indiana Department of Transportation and Purdue University, West Lafayette, Indiana, 2011. DOI: 10.5703/1288284314643

CORRESPONDING AUTHORS

Prof. Srinivas Peeta
School of Civil Engineering
Purdue University
(765) 494-2209
peeta@purdue.edu

ACKNOWLEDGMENTS

The authors acknowledge the assistance and valuable feedback from the members of the Study Advisory Committee (SAC). We would like to thank Mr. Dwayne Harris of INDOT for his helpful feedback during the entire project duration. We would also like to thank Mr. Steve Smith, Mr. Roy Nunnally and Mr. Laurence Brown from INDOT, Larry Heil from FHWA and Dr. Vince Bernardin, Jr. from BLA (Bernardin, Lochmueller & Associates, Inc.) for their valuable inputs in the development of the project. The project was funded by the Joint Transportation Research Program of Purdue University in conjunction with the Indiana Department of Transportation. We acknowledge and appreciate their support and assistance.

JOINT TRANSPORTATION RESEARCH PROGRAM

The Joint Transportation Research Program serves as a vehicle for INDOT collaboration with higher education institutions and industry in Indiana to facilitate innovation that results in continuous improvement in the planning, design, construction, operation, management and economic efficiency of the Indiana transportation infrastructure.
https://engineering.purdue.edu/JTRP/index_html

Published reports of the Joint Transportation Research Program are available at: <http://docs.lib.purdue.edu/jtrp/>

NOTICE

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views and policies of the Indiana Department of Transportation or the Federal Highway Administration. The report does not constitute a standard, specification or regulation.

This page intentionally left blank.

1. Report No. FHWA/IN/JTRP-2011/19	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle Post-Processing Techniques to Enhance Reliability of Assignment Algorithm Based Performance Measures		5. Report Date 2011	
		6. Performing Organization Code	
7. Author(s) Srinivas Peeta, Amit Kumar and Sushant Sharma		8. Performing Organization Report No. FHWA/IN/JTRP-2011/19	
9. Performing Organization Name and Address Joint Transportation Research Program Civil Engineering Building, Purdue University 550 Stadium Mall Drive West Lafayette, Indiana 47907-2051		10. Work Unit No.	
		11. Contract or Grant No. SPR-3214	
12. Sponsoring Agency Name and Address Indiana Department of Transportation State Office Building 100 North Senate Avenue Indianapolis, IN 46204		13. Type of Report and Period Covered Final Report	
		14. Sponsoring Agency Code	
15. Supplementary Notes Prepared in cooperation with the Indiana Department of Transportation.			
16. Abstract <p>This study develops an enhanced transportation planning framework by augmenting the sequential four-step planning process with post-processing techniques. The post-processing techniques are incorporated through a feedback mechanism and aim to improve the stability and convergence properties of the solution, thereby improving the reliability of the planning process. There are three building blocks of the proposed post-processing module: slope-based multi-path algorithm or SMPA, perturbation assignment, and O-D prioritization technique. SMPA is the most important part of the module and it can be used as both a post-processing algorithm and an independent static traffic assignment algorithm. In addition to SMPA, the post-processing module consists of perturbation assignment and O-D prioritization schemes. Perturbation assignment provides warm start and the O-D prioritization improves the rate of convergence by deciding the sequence in which the O-D pairs are brought into the flow update process. A detailed implementation procedure for perturbation assignment is developed along with six different criteria for O-D prioritization. Computational experiments using the Sioux Falls, Anaheim and Borman Corridor networks provide insights on these techniques. Results of the computational experiments show that SMPA has a superior rate of convergence compared to state-of-the-practice algorithms even without O-D prioritization and warm start. Results of computational experiments further reveal that a warm start using perturbation assignment and O-D prioritization provides significant benefits in improving the convergence properties and obtaining a more stable solution.</p>			
17. Key Words Static traffic assignment, Stability, Consistency, Convergence, User equilibrium		18. Distribution Statement	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 37	22. Price

This page intentionally left blank.

EXECUTIVE SUMMARY

POST-PROCESSING TECHNIQUES TO ENHANCE RELIABILITY OF ASSIGNMENT ALGORITHM BASED PERFORMANCE MEASURES

Introduction

Travel demand modeling plays a key role in the transportation system planning and evaluation process. The four-step sequential travel demand model is the most widely used technique in practice. Traffic assignment is the key step in the conventional four-step planning process. It determines the estimated traffic flow pattern in the region of interest, and hence, identifies the volumes and levels of service on the various highways/streets. The estimated performance measures are then used for strategic decision-making. The traffic assignment problem is solved by utilizing the principle of user equilibrium (UE). However, there are some key issues related to stability, consistency, and convergence of the UE assignment that trouble practitioners and planners. These problems raise questions of reliability of the performance measures that represent the assignment algorithm solution outcome.

This study addresses the three practical issues: stability, consistency, and convergence related to the traffic assignment algorithm. The problem of consistency arises due to noise in the solution at lower convergence. The study develops an enhanced transportation planning framework by augmenting the sequential four-step planning process with the post-processing techniques. The post-processing techniques are incorporated through a feedback mechanism and aim to improve the stability and convergence properties of the solution, thereby improving the reliability of the planning process. There are three building blocks of the proposed post-processing module, namely slope-based multi-path algorithm or SMPA, perturbation assignment and O-D prioritization technique. SMPA is the most important part of the module and can be used as both post-processing algorithm or as an independent static traffic assignment algorithm. In addition to SMPA, the post-processing module consists of perturbation assignment and O-D prioritization schemes. Perturbation assignment provides warm start and O-D prioritization improves the rate of convergence by deciding the sequence in which the O-D pairs are brought into flow update process.

Findings

The study findings can be separated into methodological contributions and insights from the computational experiments and data analysis. From the methodological point of view, the study formulates a static user equilibrium traffic assignment problem by decomposing the objective function of Beckmann's transformation into three parts which more rationally represents the flow update process. The study also derives the mathematical formulation of a new solution algorithm which is labeled as the slope-based multi-path algorithm or SMPA. It has better convergence characteristics compared to other potential algorithms in

practice. A hybrid approach was developed by combining the merits of simultaneous and sequential approaches to foster fast implementation of UE assignment algorithms for large-size networks and was executed in SMPA. In this approach, the shortest paths are generated and sets of paths are updated for all the O-D pairs simultaneously. Then, paths for each O-D pair are equilibrated and flows are updated based on the sequential approach. For the assignment algorithms using sequential equilibration techniques, the order in which the O-D pairs are brought into the flow update process can have significant impact on the rate of convergence and the solution stability. In this study an implementation methodology for the O-D prioritization technique was developed and six criteria for O-D prioritization were conceived and tested for a real-sized network. In addition to SMPA and the O-D prioritization technique, the technique of perturbation assignment was studied for exploiting the potential of utilizing information from previous runs of the assignment algorithm for slightly different demand or link properties. A detailed implementation procedure for perturbation assignment was also developed to facilitate seamless implementation.

Computational experiments were performed to test the effectiveness of the post-processing techniques. Results of the computational experiments reveal that the SMPA has a superior rate of convergence compared to state-of-practice algorithms. Results of computational experiments further reveal that a warm start using perturbation assignment and O-D prioritization has significant benefits over the base case of cold start and non-prioritized implementation of SMPA. These three techniques will improve the convergence characteristics of the assignment process and provide a more stable solution having lesser noise and, thereby, increasing the reliability of the planning process. The efficient use of the previous runs of the assignment process using perturbation assignment is also helpful in comparing the transportation network improvement alternatives which differ slightly (for example, an alternative involving small capacity expansions for a few links).

Implementation

The improved planning framework with the post-processing technique developed in this study will provide a better solution with less noise and a higher level of convergence compared to the conventional four-step planning process. In addition, the solution obtained by adopting this methodology will have a more stable and consistent solution and thereby will increase the reliability of the assignment process.

To facilitate seamless implementation by planning agencies, an executable code was generated for the post-processing module after proper integration of all the three techniques, namely SMPA, perturbation assignment and O-D prioritization. It is a generalized code which can be used for any network and on any computer with sufficient memory. It does not require any other software to implement this module. The developed module can be used as a post-processor or an independent traffic assignment solver. Guidelines provided with this report will help in proper formatting of the required input data files for this module.

CONTENTS

EXECUTIVE SUMMARY	i
LIST OF TABLES.....	iii
LIST OF FIGURES	iv
CHAPTER 1: INTRODUCTION.....	1
1.1 Background and Motivation.....	1
1.2 Problem Description and Statement	1
1.3 Study Objectives	2
1.4 Work Plan	2
1.5 Organization of the Report	2
CHAPTER 2: LITERATURE REVIEW.....	2
CHAPTER 3: CONCEPTUAL FRAMEWORK AND METHODOLOGY.....	4
3.1 Overview of Post-processing Framework	4
3.2 Review of Slope-Based Multi-Path Algorithm	5
3.3 Perturbation Assignment	10
3.4 O-D Prioritization Technique	12
CHAPTER 4: ANALYSIS OF DATA AND RESULTS.....	13
4.1 Computational Environment and the Test Networks	13
4.2 Relative Performance of the Algorithms	13
4.3 Evaluating the Benefits of Warm Start Using Perturbation Assignment	13
4.4 Evaluating the Benefits of O-D Prioritization Techniques	15
4.5 Relative Benefits of Proposed Techniques on Link Flow Stability	16
CHAPTER 5: EXPECTED BENEFITS, DELIVERABLES AND IMPLEMENTATION.....	20
5.1 Expected Benefits	20
5.2 Deliverables	20
5.3 Implementation and Systematic Guidelines	20
CHAPTER 6: RECOMMENDATIONS	25
6.1 Recommendations for Efficient Utilization of the Post-Processing Techniques.....	25
6.2 Recommendations for State-Wide Planning	25
CHAPTER 7: CONCLUSIONS	26
7.1 Research Summary	26
7.2 Contributions	26
7.3 Limitations of the Present Study and the Post-processing Technique	27
7.4 Future Research Directions	27
REFERENCES	27

LIST OF TABLES

Table 3.1: Four cases of perturbation assignment related to change in demand	11
Table 4.1: Deciding the optimal criteria of O-D prioritization for Borman network	17
Table 4.2a: Deciding the optimal weightage factor of O-D prioritization for Borman network	17
Table 4.2b: Deciding the optimal weightage factor of O-D prioritization for Borman network	17
Table 5.1: O-D prioritization criteria and their parametric values	22

LIST OF FIGURES

Figure 1.1: Conventional four-step travel demand modeling process	1
Figure 3.1: Conceptual flow chart of post-processing technique by feedback loop	4
Figure 3.2: Implementation flow chart of post-processing techniques	5
Figure 3.3: Sequence of steps for updating path set by simultaneous approach	7
Figure 3.4: The logic of the flow update mechanism of SMPA	7
Figure 3.5: Detailed implementation flow chart for the SMPA algorithm	11
Figure 3.6: Pseudo code for perturbation assignment	12
Figure 3.7: Deciding the criteria for O-D prioritization	12
Figure 4.1: Convergence characteristics of the algorithms for the Sioux Falls network	14
Figure 4.2: Convergence characteristics of the algorithms for the Anaheim network	14
Figure 4.3: Convergence characteristics of the algorithms for the Borman network	14
Figure 4.4: Cold-start and warm-start of SMPA for the Anaheim network with increased demand	15
Figure 4.5: Cold-start and warm-start of SMPA for the Anaheim network with decreased demand	15
Figure 4.6: Cold-start and warm-start of SMPA for the Borman network with increased demand	16
Figure 4.7: Cold-start and warm-start of SMPA for the Borman network with decreased demand	16
Figure 4.8: Combined benefits of warm-start and O-D prioritization using SMPA for the Borman network with increased demand	18
Figure 4.9: Combined benefits of warm-start and O-D prioritization using SMPA for the Borman network with decreased demand	18
Figure 4.10: Link flow stability (Link 6) for Sioux Falls network	18
Figure 4.11: Link flow stability (Link 19) for Sioux Falls network	19
Figure 4.12: Link flow stability (Link 54) for Sioux Falls network	19
Figure 4.13: Link flow stability (Link 74) for Sioux Falls network	19
Figure 5.1: Implementation methodology	21
Figure 5.2: Details of input.xls	22
Figure 5.3: Details of link.txt	23
Figure 5.4: Details of origin_nums.txt	23
Figure 5.5: Details of destin_nums.txt	24
Figure 5.6: Details of linkout.txt	24
Figure 6.1: Recommended state-wide planning model	26

CHAPTER 1: INTRODUCTION

1.1 Background and Motivation

Travel demand modeling plays a key role in the transportation system planning and evaluation process. It not only provides the basis for predicting the need for proposed transportation system improvement in addition to the size and scope of the improvement, but also forms the basis for quantifying the costs and benefits of the different alternatives of improvements (Sinha and Labi, 2007). Travel demand can be defined as the number of trips made under the given set of conditions prevalent in the transportation system. The four-step sequential travel demand model is the most widely used technique in practice for long-term planning which consists of trip generation, trip distribution, mode choice and traffic assignment. Traffic assignment is the key step in the conventional four-step planning process. It determines the estimated traffic flow pattern in the region of interest, and hence, identifies the volumes and levels of service on the various highways/streets. The estimated performance measures are then used for strategic decision-making. The traffic assignment problem (TAP) is defined as: predicting the flows on the network given the network topology, travel demand, and link performance functions.

There are two basic principles which can be employed to solve the TAP. These are known as the user equilibrium (UE) and system optimal (SO) and are based on the Wardrop's first and second principles (Wardrop, 1952), respectively. While the SO assumes that network users select paths in a coordinated manner so that total network travel time is minimized, the UE assumes that network users seek to minimize their individual travel times, which is behaviorally more realistic and, hence, mostly used in practice. UE further assumes that all users behave identically and have complete knowledge of the network traffic conditions. The UE is achieved when no user can improve his or her travel cost unilaterally by switching paths, and in this state all used paths between an origin-destination (O-D) pair have identical travel times, which is the minimum possible for that O-D pair.

Field studies (Goldfarb and Spielberg, 2005; VHB, 2006) have shown that UE assignment has lower errors, on average, when compared to other potential assignment methods (such as incremental capacity restraint, all-or-nothing, etc.) in the context of transportation project planning. However, there are some key issues related to the four-step sequential demand modeling process and UE assignment that trouble practitioners and planners. The most problematic part of the four-step model is that O-D demand acts as an input to the traffic assignment (fourth step), which is obtained after the second step (trip distribution), but trip distribution (second step) takes travel time (or travel cost) as the input, which is obtained after the fourth step. This necessitates a feedback loop going from the fourth step to the second step as shown in Figure 1.1. The feedback loop is not used in practice due to two main reasons.

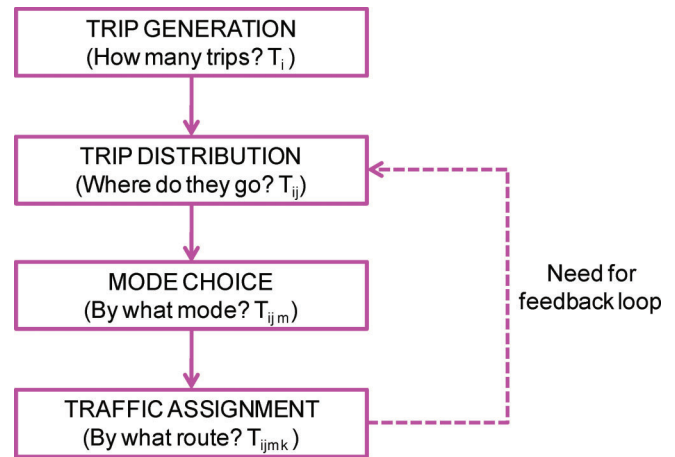


Figure 1.1 Conventional four-step travel demand modeling process

The first reason is that this loop will increase the computational time by significant amount, most of which can be attributed to the computational effort required by traffic assignment step. The second reason is that each iteration of feedback loop may results in a solution (in terms of network flows) much different than the results obtained in the previous loop, raising question when to stop the loop. In addition to the problem of interdependency of the four-step planning process mentioned above, there are three core issues related to UE assignment that trouble practitioners and planners, which are: (i) stability: solution of UE assignment is very sensitive to small changes in the network raising questions of solution stability, (ii) consistency: links far removed spatially from the alternative being studied are predicted to have significant changes in volume, and (iii) computation: solving the assignment algorithms to stable convergence requires an impractical number of iterations especially with the Frank-Wolfe (F-W) algorithm mostly used in practice.

Present study focuses on these issues related to user equilibrium traffic assignment problem (UETAP) and proposes a solution by post-processing techniques implemented through a feedback loop in the four-step sequential planning process. The core of the post-processing technique consists of an improved traffic assignment algorithm labeled Slope-Based Multi-Path Algorithm or SMPA developed by Kumar and Peeta (2010). In addition to SMPA, the post-processing technique consists of perturbation assignment and O-D prioritization schemes. Perturbation assignment provides warm start and O-D prioritization catalyzes the rate of convergence. These techniques are discussed in detail in chapter 3 of this report.

1.2 Problem Description and Statement

A recent TRB study (VHB, 2007) of small, medium and large MPOs indicates that about 75% of the small (population less than 200,000) and medium (population

between 200,000 and 1 Million) sized MPOs, and more than 90% of the large MPOs use the UE assignment method to assign highway traffic. Many MPOs used the default parameter values of their assignment software and few tracked the number of iterations required for solution convergence or examined the solution stability. However, in recent years, an increasingly-identified issue (based on in-depth analyses conducted in planning practice) with significant repercussions for practice is the problem with achieving stable convergence for the equilibrium method. Practical implementations of assignment software suggest that hundreds of iterations are required to achieve solution convergence for congested traffic networks. In many instances, this is not a feasible computational approach for real-world networks, especially when multiple runs of traffic assignment are required for comparing the different alternatives. In addition to the poor convergence rate of algorithm in practice (F-W algorithm) the solution obtained are found to be very sensitive to the input parameters. If the capacity of a link is changed slightly, the solution in terms of network flows are found to be significantly different raising concern of solution stability. This may sometimes bring inconsistency in solution outcomes, for example if a link is added or removed from a network or its capacity is changed, its effects are found to be at links very far away in the network in the UE solution based on present algorithm in practice (F-W algorithm). This is problematic, especially in the context requiring the comparison of seemingly similar planning alternatives to identify the best alternative to implement. They are problematic on two counts. First, there are questions on the reliability of the performance measures that represent the assignment algorithm solution outcome, thereby affecting the quantification of regional planning benefits of a proposed alternative. Second, the solution sensitivity raises questions of confidence in the comparison of alternatives that are similar (that is, alternatives that differ just a little in terms of the proposed network changes), so as to identify the “best” alternative to implement in practice.

1.3 Study Objectives

The focus of the present study was to investigate the merits and demerits of existing traffic assignment techniques and to develop a framework that includes an efficient traffic assignment algorithm augmented with the post-processing techniques to overcome the three key issues related to the traffic assignment techniques in practice namely: stability, consistency and convergence. The primary research objectives of the present study was to: (i) use post-processing steps/procedures to improve the stability and consistency of the equilibrium assignment solution so as to provide reliable performance measures for analyzing planning alternatives, and (ii) to investigate the performance of alternative assignment methods with the post-processing steps/procedures. The key practice objectives was

to use the research insights to provide INDOT: (i) alternative post-processing techniques to enhance the reliability of assignment-based performance measures for planning purposes, and (ii) guidelines for identifying when it is appropriate to use specific post-processing techniques based on their characteristics, strengths and limitations.

1.4 Work Plan

The existing literature focusing on the following areas was studied: equilibrium assignment models, convergence properties of the algorithms, issues related to solution stability and consistency, post-processing techniques adopted currently in the field. The merits and demerits of both conventional (Frank-Wolfe class) algorithms and recent approaches were investigated that facilitated exploring the possibility of improving the convergence properties of the algorithms. A new assignment algorithm was developed which inherits insights from the recent algorithms but its mathematical formulation is unique and is derived from the basic concepts. This algorithm can be used as an independent assignment algorithm as well as a post processor for improving the solution using warm start technique. Post-processing steps involving the new algorithm, perturbation assignment that facilitates warm start and O-D prioritization was conceptualized using the feedback loop in the four-step planning process. Then the computational experiments were carried out to test the performance of the post-processing module for the test networks.

1.5 Organization of the Report

Rest of the report is organized as follows. The next chapter gives brief overview of the developments in the domain of UE static traffic assignment. Then the next chapter gives the details of the post-processing techniques proposed in this study followed by the chapter presenting data analysis and results. And then, the expected benefits from the present study, deliverables and Implementation guidelines are presented. Finally, recommendations from the present study and conclusions are presented.

CHAPTER 2: LITERATURE REVIEW

The idea of equilibrium in traffic flows originated in 1924 when economist Frank Knight published his famous article on “Social Cost,” in which he explained how the truck operators will tend to distribute themselves between the roads if they are tolled (Knight, 1924). Still predicting the highway traffic remained an art than science until this idea was formalised in the form of two principles by John Wardrop which formed the basis for analysing traffic equilibrium and are labelled as Wardrop’s first and second principles (Wardrop, 1952). Wardrop’s first principle states that travel time (cost) of all used routes

are equal and less than those which will be experienced by a single vehicle on any unused route. The traffic flows that satisfy this principle are referred to as user equilibrium (UE) flows. According to this principle each network user tries to minimize his/her travel cost in a non-cooperative manner. As such the user equilibrium is reached when no user can lower his/her travel cost by unilateral action. Wardrop's second principle states that at equilibrium the average journey time is the minimum. This principle implies that network users behave cooperatively in selecting the routes to ensure the most efficient use of the whole network. As such some of the network users may be better placed than others who sacrifice by selecting more costly routes to make the average system cost minimum. The traffic flows that satisfy this principle are referred to as system optimal (SO) flows and can be achieved by marginal cost road pricing. The first principle defining the user equilibrium is behaviourally more realistic since each user chooses the route that is the best, and hence most widely used in practice.

Beckmann et al. (1956) proposed the user equilibrium traffic assignment problem (UETAP) as an optimization problem, which is known as Beckmann's transformation. In the same year, the linear approximation method was devised by Frank and Wolfe (1956) to solve the quadratic problem, which is known as Frank-Wolfe (F-W) algorithm. Although this method was devised to solve the quadratic optimization problem, it paved the way for obtaining the analytical solution to UETAP and was used to solve the network equilibrium problem by Bruynooghe et al. (1969), and later by LeBlanc et al. (1973, 1975) and Nguyen (1974). Florian and Nguyen (1976) and Dow and Van Vliet (1979) provided the validation studies for this method, and it was adopted as the technique for solving UETAP in commercially available software. Since then, the F-W algorithm has been the most commonly used method, which replaced the heuristic methods that were previously used in practice (e.g. capacity restraint, incremental assignment etc.).

The F-W algorithm has some advantages for practical implementation. It has a simple structure which is easy to implement. In addition, its memory requirements are low, making it computationally compatible with computers of older vintage. It also has a good convergence rate in initial iterations, but starts tailing especially as it moves closer to convergence. The main reason for its poor performance is attributed to its search direction which becomes perpendicular to the direction of maximum descent near convergence. This deficiency leads to many variants of F-W algorithm. Most of these have attempted to modify step size or search direction. Weintraub et al. (1985) suggested the use of a modified step size which is obtained by multiplying the optimal step size by some factor to reduce the zigzagging effect in the F-W algorithm. Other such examples include the methods suggested by Wolfe (1970) and Meyer (1974). Luenberger (1973) suggested an improved F-W algorithm by using the parallel tangent (PARTAN) direction

which was introduced in the TAP by LeBlanc et al. (1985) and Florian et al. (1987). Fukushima (1984) suggested a method that uses the convex combination of linear approximation solutions. In this method linear approximation solutions from previous iterations are used to generate a new search direction. Then, the directional derivatives of this new search direction and the F-W search direction are computed, and the search direction with the lower directional derivative among these two is used to generate the next updated solution. Lee and Nie (2001) proposed a method similar to Fukushima, but it uses the heuristic to decide parameters based on the congestion levels in the network for modifying the search direction. Most of these methods operate in the space of link flows and benefit from the basic structure of the F-W algorithm which requires lesser memory usage.

Over time, the need for path-based solutions was felt by planners, and led to the re-visit of the path-based method which was first proposed by Dafermos (1968) and Dafermos and Sparrow (1969). This method equilibrates a single origin-destination (O-D) pair at a time by shifting flows from the longest path to the shortest path at each move. The step size for each move is found through a linear line search, and requires enumerating paths. This makes it expensive in terms of the CPU time and memory requirements, and was hence not used. Later, evolving computing advances and resources encouraged researchers to develop path-based methods, most of which are Newton-type methods and use the second order derivative of the objective function. One such method is the disaggregate simplicial decomposition (DSD) method proposed by Larson and Patriksson (1992) which iterates between a master problem and a sub-problem. In DSD, in the initial few iterations first order reduced gradient method is employed to obtain a near-optimal solution, and then a second order diagonalized Newton method is used to determine a highly accurate solution. One important advantage of the DSD algorithm is its excellent re-optimization capability. The solution of DSD is obtained in terms of origin-destination path flow proportions which can be easily used as warm start if UETAP needs to be solved for slightly different input. Jayakrishnan et al. (1994) proposed a gradient projection (GP) algorithm to solve the UETAP based on the Goldstein-Levitin-Polyak gradient projection (GP) algorithm formulated and popularized by Bertsekas (1976, 1987). Chen et al. (2002) compared the DSD and GP algorithms taking the networks of realistic size and found that DSD required less iteration than GP but GP performed better in terms of computational time required to reach the same level of convergence. The faster rate of convergence of GP is attributed to its flow update mechanism which maintains fewer paths and obviates the need for line search for determining the step size. Bar-Gera (1999, 2002) proposed an approach which has solution variables in terms of origin-based flow proportions, and can be used to find the route flows. It makes use of an origin-based

bush structure to eliminate cycles and is conceptually similar to the destination-based algorithm of Gallager (1977) proposed for telecommunication networks. Dial (2006) proposed a path-based algorithm labelled algorithm B that also makes use of a bush-based acyclic sub-network and results in a highly efficient solution. Algorithm B uses the slopes of path costs to shift flow from the longest path to the shortest path at each move, equilibrates one O-D pair at a time in a sequential manner, and does not require line search to find the step size. Florian et al. (2009) developed an algorithm based on the projected gradient method of Rosen (1960) which also gives a solution in terms of path flows and equilibrates one O-D pair at a time sequentially. Recently, Kumar and Peeta (2010) developed an algorithm labelled the slope-based multi-path algorithm (SMPA) that uses the slope of cost function efficiently to shift flows from the set of costlier paths to the set of cheaper paths simultaneously and seeks to move path costs towards the average cost for an O-D pair at each iteration to achieve faster convergence.

CHAPTER 3: CONCEPTUAL FRAMEWORK AND METHODOLOGY

3.1 Overview of Post-processing Framework

As mentioned earlier travel demand modeling forms the basis for impact analysis of a transportation improvement projects and helps in comparing the different alternatives. In many cases different alternatives being evaluated may differ very little, but the present state of practice utilizing the sequential four-step method gives network flows which differ significantly and even unexpected to proposed improvement

in the network. This change in flow can be sometimes realistic but in many cases arises due to noise in the solution at lower level of convergence in the traffic assignment step. The amount of noise in the solution will certainly depend on the methodology adopted and how well the methodology represents the real-world phenomena. Past studies have proved that noise in the solution fades off as the higher level of convergence in the traffic assignment step is reached (Boyce et al., 2004). Hence, in practice a method will be of practical value if it provides the solution consistent with the changes in the input. Here we present a methodology which tries to overcome these demerits by post-processing techniques incorporated through a feedback mechanism in the sequential four-step modeling process. The logic of the proposed methodology has been presented by the conceptual flow chart in Figure 3.1. As depicted by this flow chart, the improved planning process starts with network topology and link performance function as the input, and four steps namely trip generation, trip distribution, modal split and traffic assignment are carried out sequentially as iteration 1. Then the results of traffic assignment in terms of paths and path flows are used by the post-processing module to obtain a more converged solution. The link flows and the link costs are then found corresponding to the highly converged solution. Link costs are returned to trip distribution step by means of feedback loop and this triggers the beginning of next iteration. Using this new link costs, a new trip matrix (O-D demand matrix) is generated by trip generation step, and then modal choice (mode split) is done as the next step. But from now onwards fourth step namely trip assignment is skipped, and network flows are obtained by the

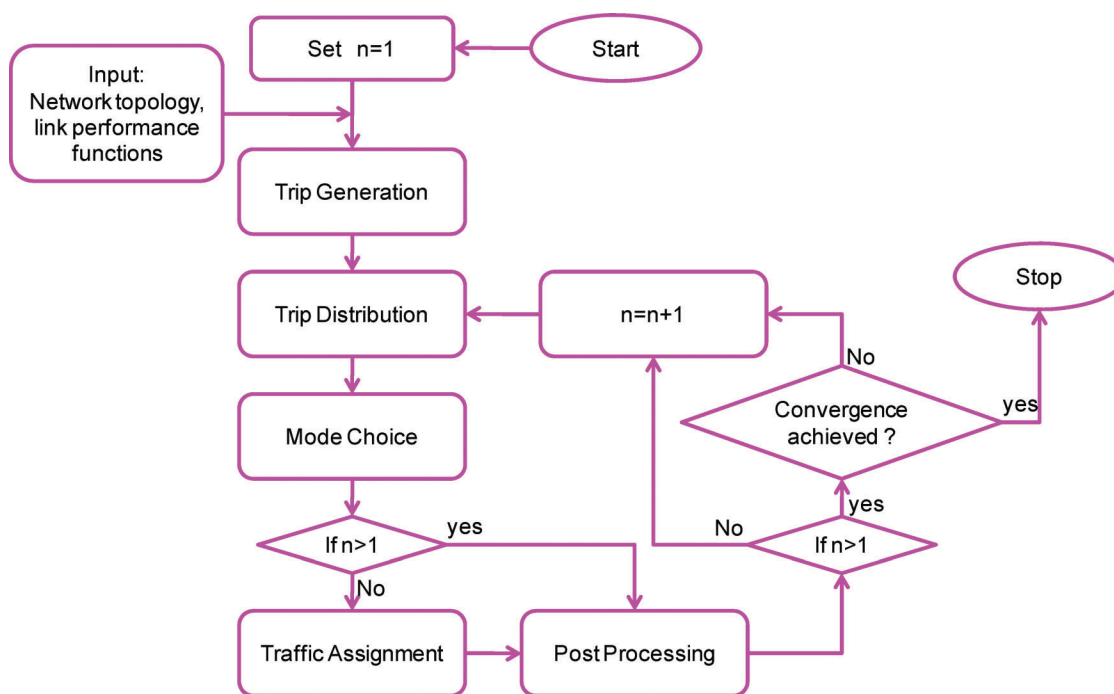


Figure 3.1 Conceptual flow chart of post-processing technique by feedback loop

post-processing module using the path flows from the previous iteration and the new trip matrix. The feedback loop involving the post-processing technique stops when the average absolute percentage change in the link flows becomes less than a threshold value.

The sequence of the steps followed in the post-processing technique has been shown by the implementation flow chart in Figure 3.2. Now, before giving details of the post-processing step, it is imperative to mention the limitation of this module. This module works only with the path based traffic assignment algorithms. There are basically two kinds of traffic assignment algorithms. The algorithms which operate in the space of link flows and discard the path information are labeled as link based algorithms, and the algorithms that operate in the space of path flows and retain the path information are labeled as path based algorithm. As clear from the Figure 3.2, proposed post-processing methodology takes paths and path flows as the input and hence will not work if paths and path flows are not available as in the case of link based traffic assignment algorithms.

As shown in Figure 3.2, there are three building blocks of the post-processing technique namely, perturbation assignment, O-D prioritization and SMPA. The SMPA is the most important part of the post-processing technique, hence a brief review of this algorithm is provided first (for details see Kumar and Peeta, 2010) and then the details of perturbation assignment and O-D prioritization techniques are presented.

3.2 Review of Slope-Based Multi-Path Algorithm

3.2.1 Preliminaries

The slope-based multi-path algorithm (SMPA) is based on the sequential decomposition technique where O-D pairs are equilibrated one at a time in sequential order. It operates in the space of path flows and updates all feasible paths of an O-D pair simultaneously. The novelty of this algorithm lies in two aspects of the flow update mechanism: (i) obviating the need for a line search in each iteration, and (ii) the way in which the sensitivity of path costs relative to flow, referred to as slopes, are used in the equilibration process.

The SMPA algorithm inherits some insights from gradient projection (GP) algorithm of Jayakrishnan et al. (1994), Algorithm B of Dial (2006) and New GP (FGP) of Florian et al. (2009), but its flow update mechanism is new and differs from them. Akin to the GP algorithm of Jayakrishnan et al., the SMPA algorithm operates in the path flow space and uses a projection to the constraint boundary by making negative flow variables zero whenever the minimum in the search direction violates the non-negativity constraints. The difference lies in the search direction; Jayakrishnan et al. use the cost of the cheapest path to determine the search direction while the SMPA uses the average cost for it. Thereby, the SMPA algorithm seeks to move path costs towards the average cost for an O-D pair at each iteration, akin to Florian et al. (2009).

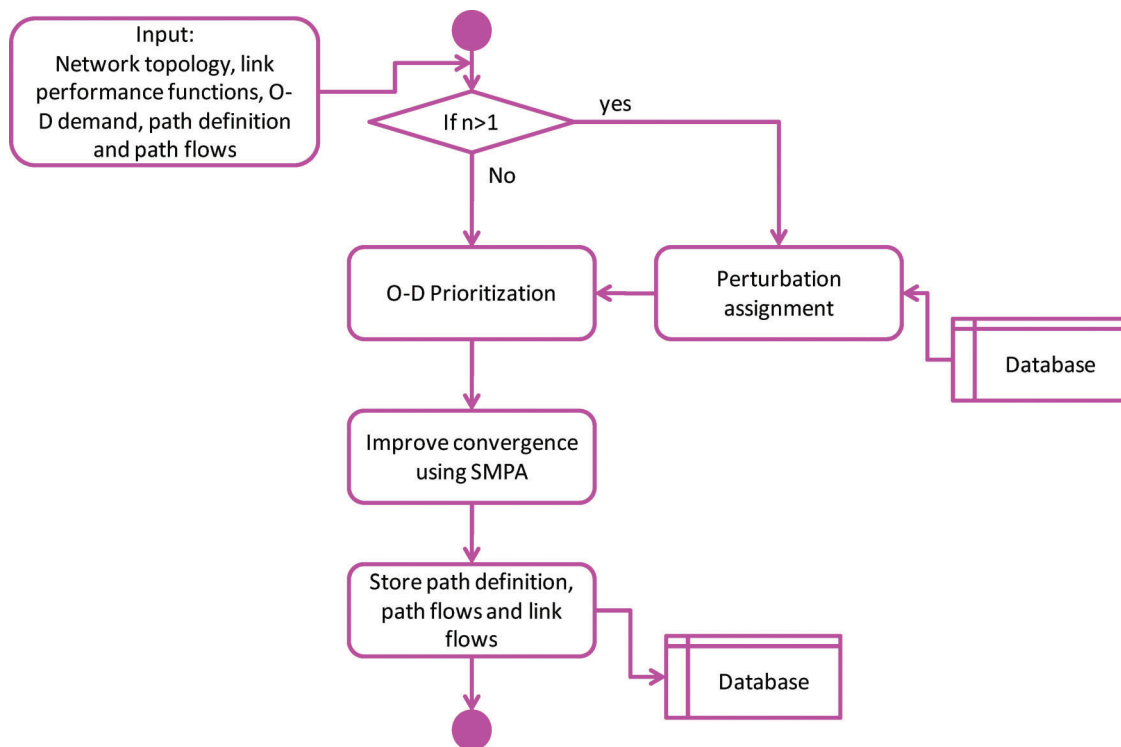


Figure 3.2 Implementation flow chart of post-processing techniques

However unlike Florian et al., it does not require a line search for the move size determination. Further, the SMPA has different flow update mechanisms for the set of paths with higher costs than average as compared to the set of paths with costs lower than average. Finally, the SMPA algorithm inherits the concept of equilibrating an O-D pair before bringing another O-D pair into the equilibration process, similar to Dial's algorithm B. However, it operates on all paths of an O-D pair simultaneously unlike in Dial's algorithm B which operates on just a pair of paths at a time.

3.2.2 Steps of SMPA

The SMPA algorithm consists of an inner loop which seeks the equilibration of an O-D pair and an outer loop that sequentially moves from one O-D pair to the next and checks termination criteria after all O-D pairs are considered. SMPA adopts the termination criteria in terms of Normalized gap ($Ngap$) or average excess cost (for details of $Ngap$ see Rose et al., 1988). At each iteration, inner loop of SMPA equilibrates all the paths in the path set between an O-D pair simultaneously. The paths in the set are divided into two subsets- the cheaper path set having travel cost lesser than the average travel cost for the O-D pair, and costlier path set having cost greater than average travel cost for the O-D pair. Then flows are shifted from the set of costlier paths to the set of cheaper paths so that their costs are brought towards the average cost. In this flow update process slopes of cost functions are utilized. The steps of the SMPA algorithm are as below:

Step 1:	Initialize the network using an all-or-nothing (AON) assignment or a warm start. Update the link flows, link costs, slopes of cost functions, and path costs.
Step 2:	Check the termination criteria. If termination criteria are satisfied, then stop; the UE solution is the set of the path flows and path costs. Else, go to Step 3.
Step 3:	Select the first O-D pair in the sequence.
Step 4:	Find the average cost (c_{av}) of paths in the feasible set (having non-zero flows) for the O-D pair. Then, find the shortest path. If this path is not present in the set of feasible paths then include it in the set of feasible paths and update c_{av} . Else, go to Step 5.
Step 5:	Update flows for paths having cost greater than c_{av} .
Step 6:	Update flows for paths having cost lower than c_{av} satisfying O-D demand.
Step 7:	Update link flows, link costs, slopes of cost functions, and path costs.
Step 8:	If the current O-D pair is equilibrated, go to Step 9. Else, go to Step 4.
Step 9:	If this is the last O-D pair, go to Step 2. Else, select next O-D pair and go to Step 4.

3.2.3 The Need for a Hybrid Approach

The sequential decomposition based (one at a time based) technique is good for small and medium size networks but it is not good for very large size networks

like large city or state-wide network. The reason is: gain in higher rate of convergence per iteration is compensated by the computational time required for generating the shortest paths and updating the path set based on sequential approach. Hence a hybrid version of SMPA was developed in which shortest paths are generated and set of paths are updated for all the O-D pairs simultaneously and then paths for each O-D pair are equilibrated and flows are updated based on the sequential approach. Hence step 2 and 4 of the SMPA are modified as below keeping other steps unchanged:

Step 2:	Check the termination criteria. If termination criteria are satisfied, then stop; the UE solution is the set of the path flows and path costs. Else, update the path sets of all the O-D pairs based on the simultaneous approach and go to step 3.
Step 4:	Find the average cost (c_{av}) of paths in the feasible set (having non-zero flows) for the O-D pair.

3.2.4 Updating the Path Sets Based on Simultaneous Approach

The sequence of the steps followed in generating the shortest paths and updating the path sets for all the O-D pairs based on simultaneous approach is shown in Figure 3.3. As evident from the flow chart shown in this figure the shortest paths are generated for all destinations originating from a single origin at a time and then generated paths are compared with the existing path set and added only if it is not present in the path set. While doing so those destinations are skipped for which the travel demand from the present origin is zero. Once the path set for all the designations for the present origin are updated then process shifts to the next origin. The process continues till all origins are covered once.

3.2.5 Flow Update Mechanism of SMPA

To understand the flow update mechanism of SMPA let us consider an intermediate iteration of the equilibration process where a particular O-D pair is being equilibrated. Here, the word intermediate stage signifies that the path flows do not satisfy UE. As the UE conditions are not satisfied, paths with non-zero flows will not have equal cost for an O-D pair. This implies the existence of paths with non-zero flows but with costs higher than that of the minimum cost path for that O-D pair. Then, the equilibration process will entail the shifting of flows from the costlier to the cheaper paths. For example, let us assume that the O-D pair in equilibration process has six paths in the feasible set of paths labeled 1 to 6 as shown in the Figure 3.4. The paths labeled 1, 2 and 3 have costs lesser than the average cost and will form the cheaper path set. The paths labeled 4, 5 and 6 have costs greater than the average cost and will form the costlier path set. Then, the equilibration process will entail the shifting of flows from the costlier to the cheaper paths. An optimal shift

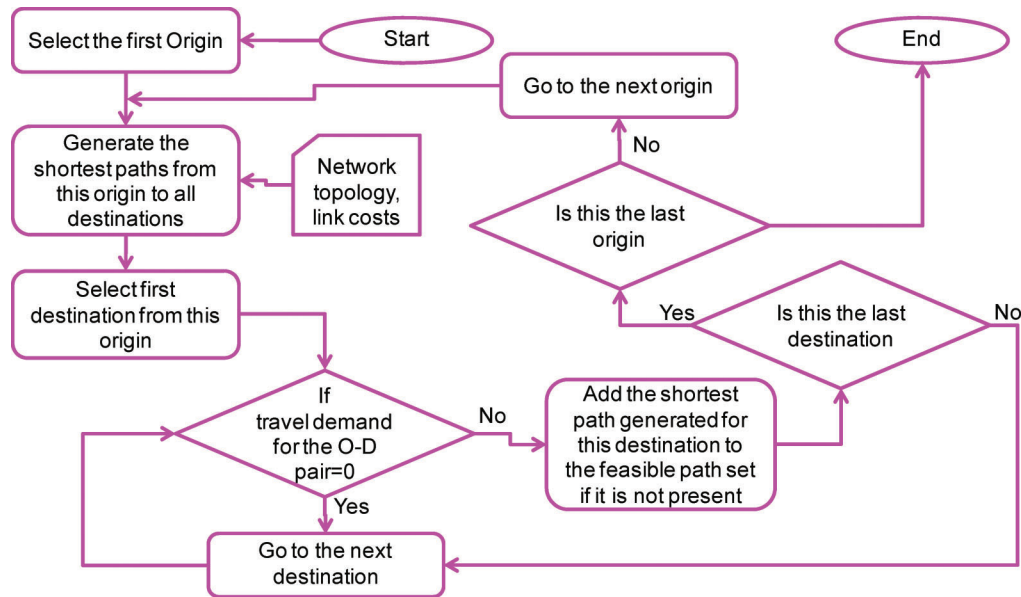


Figure 3.3 Sequence of steps for updating path set by simultaneous approach

from the costlier paths to the cheaper paths will move the resulting network flows towards equilibrium. While shifting flows, the flow update mechanism of SMPA seeks to reduce the costs of costlier paths and bring them to the average cost (c_{av}) for the O-D pair, and aims to increase the costs of the cheaper paths to a value μ . The positions of the paths after the flow update are represented by the dotted ellipses in the figure. After each move, the flows, the costs and the slopes of links and paths are updated. The new average path cost for the O-D pair is found based on the updated flows. This process is repeated iteratively whereby the difference

between c_{av} and μ decreases in each successive iteration. The O-D pair is assumed to be equilibrated if the difference between its μ and c_{av} becomes less than a pre-specified small threshold value β . Once an O-D pair is equilibrated, the next O-D pair is brought into the equilibration process in a sequential manner.

3.2.6 Problem Formulation and Algorithm Development

Let us consider a network with a set of nodes \mathcal{N} , and a set of links \mathcal{A} . Let \mathcal{R} be the set of origin nodes and \mathcal{S} be

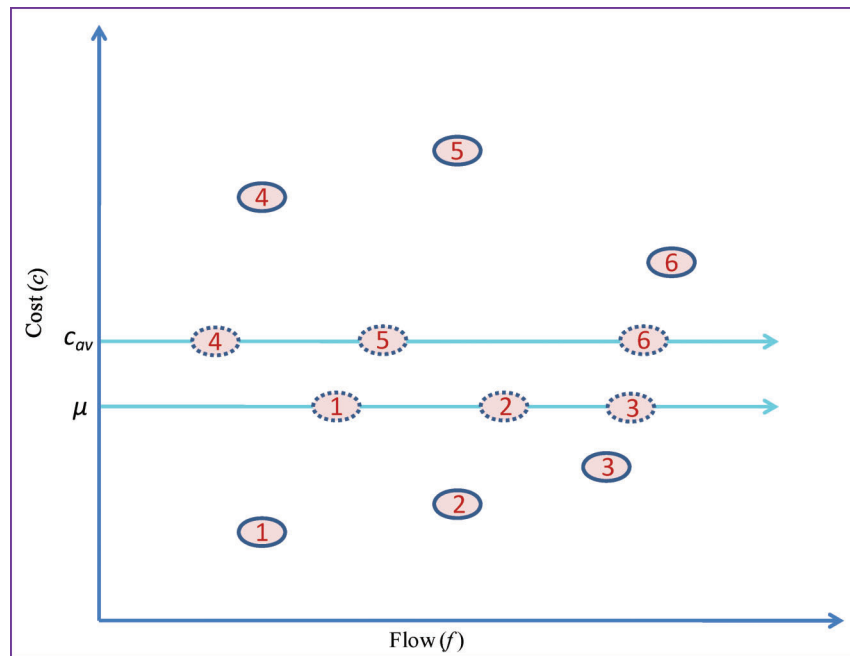


Figure 3.4 The logic of the flow update mechanism of SMPA

the set of destination nodes which may not be mutually exclusive. The set of paths that connect an O-D pair $r-s$ is denoted as K^{rs} and the O-D demand as q_{rs} . Let x_a and c_a represent the flow and cost on link a respectively, and f_k^{rs} and c_k^{rs} represent the flow and cost on path k of O-D pair $r-s$, respectively. The cost of traveling on a path is equal to the sum of the costs on the links in that path, where the link cost is a function $c_a(x_a)$ of the flow on that link. The UE assignment problem can be formulated as an optimization problem by the well-known Beckmann's transformation (Beckmann et al., 1956):

$$\min \quad Z(x) = \sum_{a \in A} \int_0^{x_a} c_a(w) dw \quad (1)$$

Subject to:

$$\sum_k f_k^{rs} = q_{rs}, \quad \forall rs \text{ (flow conservation constraint)} \quad (1a)$$

$$f_k^{rs} \geq 0, \quad \forall k, \quad \forall rs \text{ (non-negativity constraint)} \quad (1b)$$

$$x_a = \sum_r \sum_s \sum_k f_k^{rs} \delta_{a,k}^{rs}, \quad (1c)$$

$\delta_{a,k}^{rs} = 1$ when link a lies on path k , 0 otherwise

The equivalency of this formulation to the static UE problem and the uniqueness of the solution in terms of link flows is based on the following: (i) the O-D demand is constant and non-negative for all O-D pairs, (ii) the link costs are positive and the link cost functions are monotonically increasing, continuously differentiable functions of flow, and (iii) a link cost depends only on the flow on that link and does not depend on the flow on other links.

The above formulation is usually solved using an iterative approach. The approach starts with a feasible point and seeks to bring it closer to equilibrium in successive iterations by identifying an optimal search direction and move size. Let us consider an intermediate iteration of the equilibration process by SMPA, represented by feasible links flows that do not satisfy UE. Here, feasible flows refer to the vector of flows that satisfies the set of constraints (1a, 1b, and 1c) in the UE formulation. As the UE conditions are not satisfied, paths with non-zero flows will not have equal cost for an O-D pair. This implies the existence of paths with non-zero flow but with costs higher than that of the minimum cost path for that O-D pair. Then, the equilibration process will entail the shifting of flows from the costlier to the cheaper paths. An optimal shift from the costlier paths to the cheaper paths will move the resulting network flows towards equilibrium, thereby decreasing the objective function value. Dial (2006) proposed a mechanism for the optimal flow transfer from the costlier path to the cheaper path, by

taking just one pair of paths at a time. Akin to algorithm B, flow update mechanism of SMPA equilibrates the entire feasible set of paths belonging to an O-D pair simultaneously. That is, instead of transferring flow from one path (costlier) to another path (cheaper), flow is optimally shifted from the set of paths with higher path costs than average for that O-D pair to the set of paths with lower costs than average, inheriting the idea from the GP algorithm of Florian et al. (2009). To enable this in the SMPA, the objective function during the equilibration process for an O-D pair $r-s$ can be decomposed into three parts:

$$\begin{aligned} f(x) = & \sum_{a \in \bar{P}^{rs} \cup \underline{P}^{rs}} \int_0^{x_a - \bar{\Delta}x_a} c_a(w) dw \\ & + \sum_{a \in \bar{P}^{rs}} \int_0^{x_a + (\bar{\Delta}x_a - \bar{\Delta}x_a)} c_a(w) dw \\ & + \sum_{a \in A \setminus (\bar{P}^{rs} \cup \underline{P}^{rs})} \int_0^{x_a} c_a(w) dw \end{aligned} \quad (2)$$

Where,

\bar{P}^{rs} = set of costlier paths, comprising of paths having cost greater than the average cost for the O-D pair $r-s$

\underline{P}^{rs} = set of cheaper paths, comprising of paths having cost lesser than the average cost for the O-D pair $r-s$

$\bar{\Delta}x_a$ = change in flow for link a due to flow update of paths in the costlier set

$\underline{\Delta}x_a$ = change in flow for link a due to flow update of paths in the cheaper set

The third part in Equation (2) considers links which do not belong to any path in the feasible set for the O-D pair $r-s$ being equilibrated. The equilibration process represented by Equation (2) is repeated until all O-D pairs are equilibrated. The superscript rs is dropped hereafter for simplicity of notation as we focus on an O-D pair $r-s$.

Path set \bar{P} (set of costlier paths)

For path set \bar{P} , we transfer flow such that the new path costs resulting from the new flows bring the path costs towards the average cost across all paths for the O-D pair $r-s$. Using a first order Taylor expansion:

$$\bar{c}_k(\bar{f}_k - \Delta \bar{f}_k) = \bar{c}_k(\bar{f}_k) - \bar{s}_k(\bar{f}_k) \cdot \Delta \bar{f}_k = c_{av}, \quad \forall k \in \bar{P} \in K \quad (3)$$

Where,

\bar{c}_k = cost of path k belonging to the set \bar{P}

\bar{f}_k = flow of path k belonging to the set \bar{P}

\bar{s}_k = first derivative of the cost function of path k belonging to the set \bar{P}

c_{av} = average cost obtained by averaging the feasible path costs for the O-D pair

From Equation (3) we obtain:

$$\Delta \bar{f}_k = \frac{\bar{c}_k(\bar{f}_k) - c_{av}}{\bar{s}_k(\bar{f}_k)} \quad (4)$$

Path set \underline{P} (set of cheaper paths)

For path set \underline{P} , we add flow to the paths such that the new path costs resulting from the new flows bring these costs to a higher value μ . This new cost μ may not be equal to average path cost c_{av} for O-D pair $r-s$, but will definitely be close to it. Using a first order Taylor expansion:

$$c_l(\underline{f}_l + \Delta \underline{f}_l) = c_l(\underline{f}_l) + s_l(\underline{f}_l) \cdot \Delta \underline{f}_l = \mu, \quad \forall l \in \underline{P} \in K \quad (5)$$

Where,

c_l = cost of path l belonging to the set \underline{P}
 \underline{f}_l = flow of path l belonging to the set \underline{P}
 s_l = first derivative of the cost function of path l belonging to the set \underline{P}

From Equation (5) we obtain:

$$\Delta \underline{f}_l = \frac{\mu - c_l(\underline{f}_l)}{s_l(\underline{f}_l)} \quad (6)$$

For flow conservation for the O-D pair, the total flow transferred from the various paths from set \bar{P} should be equal to the sum of flows being added to the paths of set \underline{P} . Hence,

$$\sum_{k \in \bar{P}} \Delta \bar{f}_k = \sum_{l \in \underline{P}} \Delta \underline{f}_l \quad (7)$$

Using Equations (4), (6) and (7) we obtain:

$$\begin{aligned} \sum_{k \in \bar{P}} \frac{\bar{c}_k(\bar{f}_k) - c_{av}}{\bar{s}_k(\bar{f}_k)} &= \sum_{l \in \underline{P}} \frac{\mu - c_l(\underline{f}_l)}{s_l(\underline{f}_l)} \\ \Rightarrow \sum_{k \in \bar{P}} \frac{\bar{c}_k(\bar{f}_k)}{\bar{s}_k(\bar{f}_k)} - \sum_{k \in \bar{P}} \frac{c_{av}}{\bar{s}_k(\bar{f}_k)} &= \mu \sum_{l \in \underline{P}} \frac{1}{s_l(\underline{f}_l)} - \sum_{l \in \underline{P}} \frac{c_l(\underline{f}_l)}{s_l(\underline{f}_l)} \\ \Rightarrow \mu &= \frac{\sum_{k \in \bar{P}} \frac{\bar{c}_k(\bar{f}_k)}{\bar{s}_k(\bar{f}_k)} - \sum_{k \in \bar{P}} \frac{c_{av}}{\bar{s}_k(\bar{f}_k)} + \sum_{l \in \underline{P}} \frac{c_l(\underline{f}_l)}{s_l(\underline{f}_l)}}{\sum_{l \in \underline{P}} \frac{1}{s_l(\underline{f}_l)}} \end{aligned}$$

From Equation (6) we have:

$$\Delta \underline{f}_l = \frac{\mu - c_l(\underline{f}_l)}{s_l(\underline{f}_l)} = \frac{\mu}{s_l(\underline{f}_l)} - \frac{c_l(\underline{f}_l)}{s_l(\underline{f}_l)}$$

Substituting for μ ,

$$\Delta \underline{f}_l = \frac{\sum_{k \in \bar{P}} \frac{\bar{c}_k(\bar{f}_k)}{\bar{s}_k(\bar{f}_k)} - \sum_{k \in \bar{P}} \frac{c_{av}}{\bar{s}_k(\bar{f}_k)} + \sum_{l \in \underline{P}} \frac{c_l(\underline{f}_l)}{s_l(\underline{f}_l)}}{\frac{s_l(\underline{f}_l) \cdot \sum_{l \in \underline{P}} \frac{1}{s_l(\underline{f}_l)}}{s_l(\underline{f}_l)}} - \frac{c_l(\underline{f}_l)}{s_l(\underline{f}_l)}$$

We now have the following flow update mechanism:

$$\bar{f}_k \rightarrow \bar{f}_k - \Delta \bar{f}_k, \quad k \in \bar{P} \quad (A)$$

$$\underline{f}_l \rightarrow \underline{f}_l + \Delta \underline{f}_l, \quad l \in \underline{P} \quad (B)$$

Expressions (A) and (B) are used iteratively until the maximum difference between the costs of used paths (that is, those with non-zero flow) is less than a predefined threshold value β . While transferring flow at each move, the maximum amount of flow which can be transferred from any path is at most equal to the current flow on that path to satisfy the non-negativity constraint. Hence, the $\Delta \bar{f}_k$ and $\Delta \underline{f}_l$ to be used at each move are:

$$\Delta \bar{f}_k = \min \left\{ \bar{f}_k, \frac{\bar{c}_k(\bar{f}_k) - c_{av}}{\bar{s}_k(\bar{f}_k)} \right\} \quad (8)$$

$$\Delta \underline{f}_l = \frac{\sum_{k \in \bar{P}} \min \left\{ \bar{f}_k, \frac{\bar{c}_k(\bar{f}_k) - c_{av}}{\bar{s}_k(\bar{f}_k)} \right\} + \sum_{l \in \underline{P}} \frac{c_l(\underline{f}_l)}{s_l(\underline{f}_l)}}{\frac{s_l(\underline{f}_l) \cdot \sum_{l \in \underline{P}} \frac{1}{s_l(\underline{f}_l)}}{s_l(\underline{f}_l)}} - \frac{c_l(\underline{f}_l)}{s_l(\underline{f}_l)} \quad (9)$$

Analysis of Convergence: The flow vector which solves the minimization problem represented by Beckmann's transformation (Equations 1, 1a, 1b, and 1c) also satisfies the user equilibrium conditions (Sheffi, 1985). This implies that the minimum of the objective function (1) is obtained when all used paths between an O-D pair have equal (and lowest) cost. Since the decomposed objective function (2) is equivalent to the objective function (1), its minimum will also be achieved when all paths in the feasible set have equal cost. In the decomposed objective function, the links that belong to the third part do not participate in the equilibration process. Hence, the minimum of $f(x)$ is obtained when the sum of the first two terms is the minimum. As the true equilibrium for real networks is

difficult to reach, the equilibrium is assumed to be reached when the difference between path costs in the feasible set is less than a pre-specified threshold β . The equilibration process represented by the flow update using Expressions (A) and (B) iteratively continues until the difference between c_{av} and μ becomes less than β , at which point it is terminated. Then, to prove convergence, it is sufficient to prove that the flow update mechanism decreases the value of the objective function.

Proof of convergence of SMPA by contradiction: The flow update mechanism of SMPA shift flows from costlier paths to cheaper paths. The flow shift is terminated when difference between their costs becomes less than the threshold parameter β ($\beta > 0$). Let us suppose that the flow update process increases the value of the objective function at the termination of the equilibration process. This would imply that the increase in the second part of the objective function is larger than the decrease in first part. This implies that the increase to the objective function value due to the flow increases in the cheaper path set is more than the decrease to the objective function value due to the flow decreases in the costlier path set. This indicates that at the termination point, the flow shifts have turned the costs of cheaper paths higher than the costs of costlier paths, which contradicts our previous assumption that flow shift process is stopped when path cost difference becomes less than β . This completes the proof.

3.2.7 Practical Aspects of the SMPA

The description of the SMPA algorithm in the previous section seems to suggest that it satisfies all constraints in each iteration. However, a careful inspection of the move direction for the cheaper paths indicates the possibility of a theoretical violation of the non-negativity constraint. This situation may arise when the second term in equation (9) is greater than the first term, resulting in a negative flow change for the cheaper path set. Then, if this flow change is greater than the existing path flow, it will result in a negative path flow for that path. To visualize this, consider a situation in which the move direction is such that the value of μ is below the average cost c_{av} , and the feasible set has a path with cost smaller than the average cost but greater than μ . Since the SMPA move direction tends to bring all paths having cost lesser than the average cost towards μ , if a path cost is greater than μ , the move direction will seek to reduce it by decreasing its flow. While flow conservation is satisfied in this process, the possibility exists that this move may result in an infeasible flow, thereby violating the non-negativity constraint. While such a situation is not common and will involve small amounts of flow, it is important to resolve this theoretical issue to make the flow update mechanism consistent with the formulation constraints. This is done by the backtracking in the direction opposite to the move direction for paths in the cheaper path set up to the point at which the sum of

the flow changes becomes equal to the magnitude of the negative flow. Thereby, the infeasible flow is projected back to the feasible space, satisfying the non-negativity constraint and the conservation of flow. The detailed sequence of the steps to implement the SMPA algorithm is illustrated in Figure 3.5.

The second aspect which needs discussion is the interdependencies among path costs. Some paths in the feasible set may have some common links; then, the increase or decrease of flow on one path will alter the cost on the other path(s). The SMPA addresses this problem by incorporating slope (of path cost) terms in the move direction. As the slope of a path cost is obtained by summing the slopes of the associated link cost functions, if a link appears in more than one path the slope of that link will appear the corresponding number of times in the flow update equation of the cheaper path set thereby decreasing the magnitude of the move direction.

The third aspect is the notion of introducing a scaling factor α . The scaling factor here refers to a parameter which is used to scale the move size. It is used to speed up the convergence, and for the SMPA it can be any positive number which remains constant for all iterations. A scaling factor is introduced in the move direction for the costlier path set. As the move direction for the cheaper path set is derived by satisfying the flow conservation constraint (Equation 7), it is automatically introduced in its expression. The move directions after introducing the scaling factor are shown in Equations (10) and (11); they replace Equations (8) and (9) during the implementation of the SMPA algorithm. The value of scaling factor needs to be calibrated before using the SMPA. Its optimal value will vary from one network to another.

$$\Delta \bar{f}_k = \min \left\{ \bar{f}_{k,a} \frac{\bar{c}_k(\bar{f}_k) - c_{av}}{\bar{s}_k(\bar{f}_k)} \right\} \quad (10)$$

$$\Delta \underline{f}_l = \frac{\sum_{k \in \bar{P}} \min \left\{ \bar{f}_{k,a} \frac{\bar{c}_k(\bar{f}_k) - c_{av}}{\bar{s}_k(\bar{f}_k)} \right\} + \sum_{l \in P} \frac{\underline{c}_l(\underline{f}_l)}{\underline{s}_l(\underline{f}_l)}}{\underline{s}_l(\underline{f}_l) \cdot \sum_{l \in P} \frac{1}{\underline{s}_l(\underline{f}_l)}} - \frac{\underline{c}_l(\underline{f}_l)}{\underline{s}_l(\underline{f}_l)} \quad (11)$$

3.3 Perturbation Assignment

Perturbation assignment is the other important building block of the post-processing module. It provides the warm start to the SMPA. This technique is useful when traffic assignment needs to be carried out with a slightly different input (Kupsizewska and Vliet, 1998). This small change in input can be in three

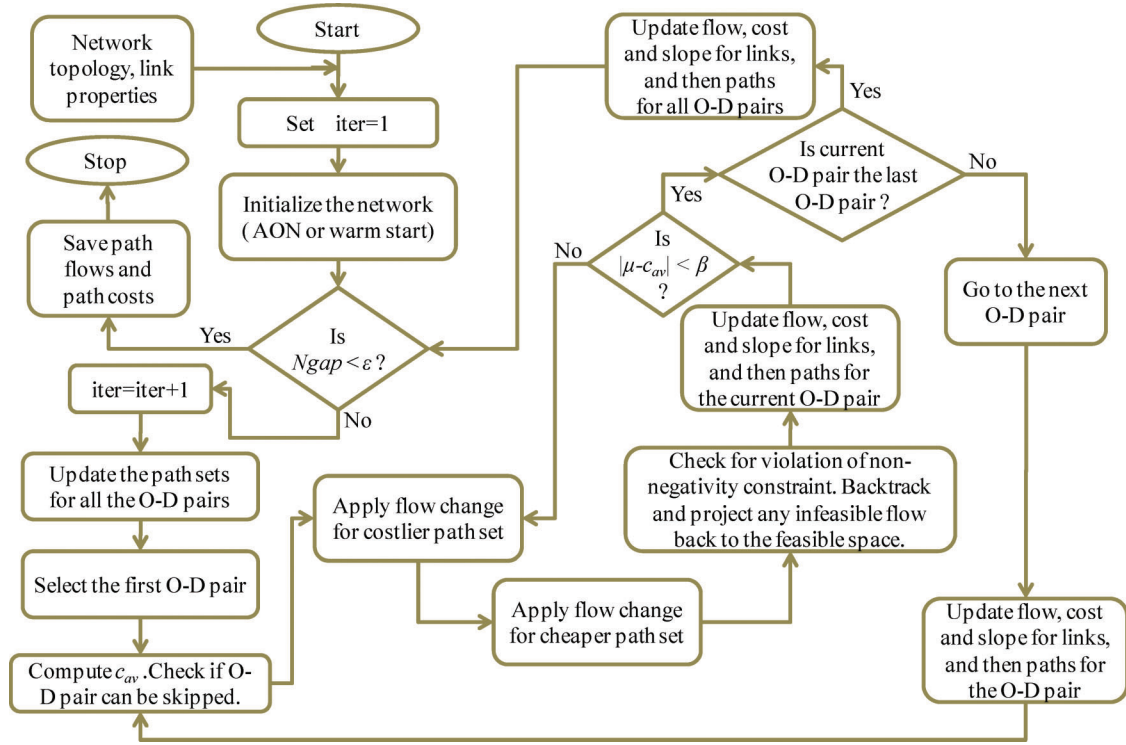


Figure 3.5 Detailed implementation flow chart for the SMPA algorithm

ways- when O-D demand changes, when network topology changes and when link properties changes for some of the links. But perturbation assignment is mainly useful when O-D demand changes by a small amount. When network topology changes, new efficient paths may come in the path set and some paths in the path set may get deleted. In the case where a new path is generated, the flow shift will be done by SMPA. But in the case when a path is deleted its flow is shifted to the rest of the paths in the path set equally. When the link properties change the flow update is mostly done by SMPA. But, for the proposed post-processing module implemented through feedback mechanism, O-D demand is most likely to change from one feedback loop to another especially in the initial loops, provided the free flow condition does not exist for most part of the network. Four cases can arise with respect to the change in the O-D demands from one feedback loop to another which is summarized in table 3.1.

Although the case 4 is most important from the implementation point of view, implementation strategies for perturbation assignment for all the four

possible cases were devised carefully to maintain the feasibility of solution space and its consistency with the input data, and are presented below:

Implementation strategy for case 1:

- Skip this O-D pair and go to next O-D pair

Implementation strategy for case 2:

- Generate the shortest path
- Assign the total demand to the shortest path

Implementation strategy for case 3:

- Delete the path set for this O-D pair along with the path flows

Implementation strategy for case 4:

- Calculate the flow proportion using the following expression:

$$\eta_k^{rs} = \frac{f_k^{rs}}{\sum_k f_k^{rs}}$$

where,

η_k^{rs} = Flow proportion for path k for O – Dpair r – s

f_k^{rs} = Path flow for path k for O – Dpair r – s

- Calculate the new path flows as below:

New path flow = flow proportion X new demand

The mechanism used for the perturbation assignment is shown by the pseudo code in Figure 3.6. As evident

TABLE 3.1
Four cases of perturbation assignment related to change in demand

Case No.	Old O-D demand	New O-D demand
1	Zero	Zero
2	Zero	Non-zero
3	Non-zero	Zero
4	Non-zero	Non-zero

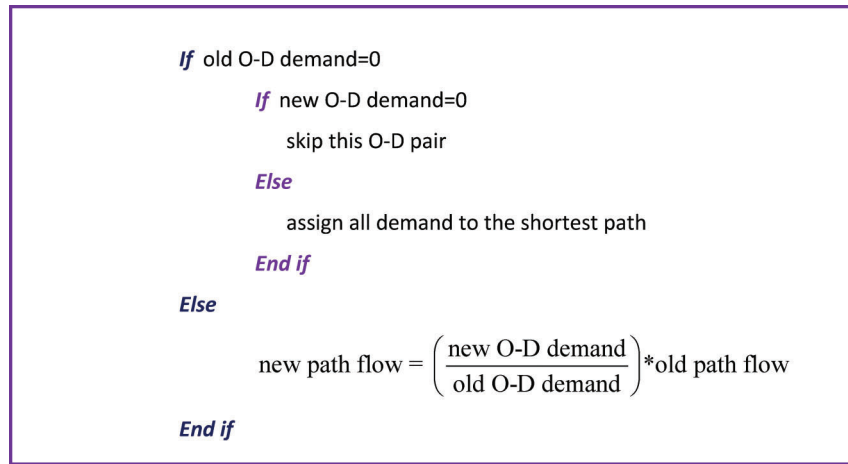


Figure 3.6 Pseudo code for perturbation assignment

from the figure, if both old and new O-D demand is zero then that O-D pair is skipped. If the old O-D demand is zero but the new O-D demand is non-zero then complete demand is assigned to the shortest path. But in the case, where both new and old O-D demands are non-zero for an O-D pair, path flows for that O-D pair is changed in the same proportion as the change in the demand.

Limitations of perturbation assignment

1. Most binding limitation of this approach is that it can be used only with the path-based traffic assignment algorithms.
2. This approach will become less beneficial if the network conditions changes by a significant amount. For example if the O-D demand or network topology changes significantly, this method can become less beneficial. But this limitation is less restrictive as still this method can be used.

3.4 O-D Prioritization Technique

There are two types of path-based traffic assignment algorithms. One that equilibrates all the O-D pairs simultaneously is termed as all-at-a-time algorithm, and the algorithm that equilibrates different O-D pairs one at a time in some sequence is termed one-at-a-time algorithm. The SMPA comes under the second category. For this kind of algorithm the sequence in which different O-D pairs are brought into equilibration process determines the rate of convergence of the algorithm. The aim of O-D prioritization is to determine the sequence which enables the algorithm to achieve faster and stable convergence. Figure 3.7 shows the steps needed to identify the best criteria for O-D prioritization. For every network there can be multiple possible ways of O-D prioritization and we need to find out which is the best for the given network. At this point it is important to mention that the best

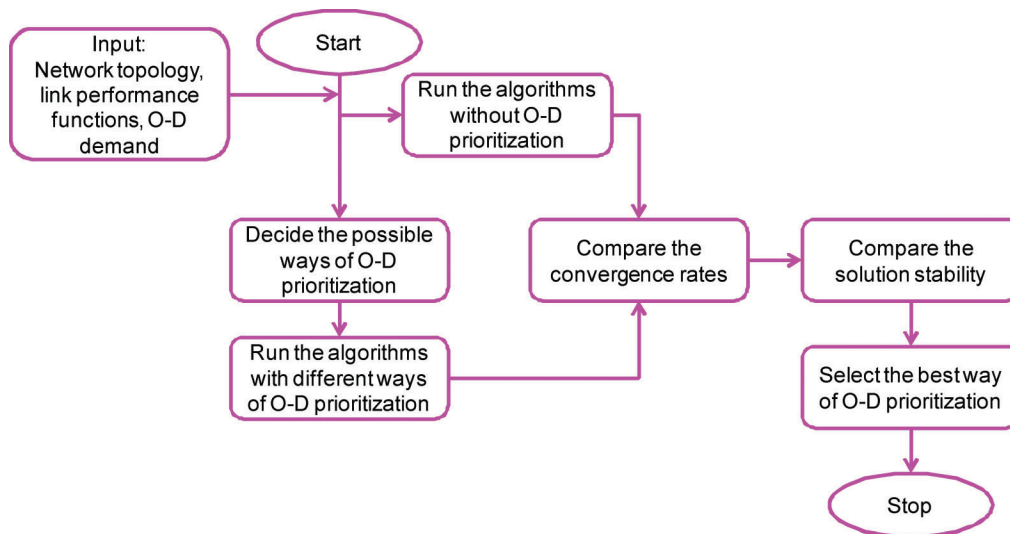


Figure 3.7 Deciding the criteria for O-D prioritization

way of O-D prioritization can be different for different network and different algorithms. But it needs to be carried out just once for a given network.

Higher order O-D demands are likely to create higher impact on traffic conditions compared to lower ones. Based on this philosophy two criteria for O-D prioritization can be generated, namely ascending or descending order of O-D demand. Because in reality when drivers are shifting routes, the question comes who gets closer to equilibrium first, the drivers having destination closer to origin or distant ones? The free flow travel time can act as the proxy for distance between the different origins and destinations. Hence, ascending and descending orders of free flow travel time has been adopted as another potential criterion for O-D prioritization in this project. Since each of those O-D demand and free flow travel time has some meaningful effect on the traffic conditions, it is appealing to simultaneously look at those two factors after giving a relative weight to them. This formed another possible way of O-D prioritization. Hence, following six O-D prioritization schemes have been tested in this project:

1. Ascending order of O-D demand (represented as $ODPr=1$)
2. Descending order of O-D demand (represented as $ODPr=-1$)
3. Ascending order of free flow travel time (represented as $ODPr=2$)
4. Descending order of free flow travel time (represented as $ODPr=-2$)
5. Ascending order of weighted measure of O-D demand and free flow travel time (represented as $ODPr=3$)
6. Descending order of weighted measure of O-D demand and free flow travel time (represented as $ODPr=-3$)

For the fifth and sixth criteria of O-D prioritization ($ODPr=3$ and -3) mentioned above, the priority index is calculated based on the parameter “weightage factor” as below:

- Normalized weight is calculated as

$$nwt = \text{mean}[\text{demand}] / \text{mean}[\text{free flow travel time}]$$

- Normalized relative weight is calculated as

$$nwtr = nwt * \text{weightage factor}$$

- priority index vector is then calculated as

$$[\text{priority index vector}] = [\text{demand vector}] + nwtr * [\text{vector of free flow travel time}]$$

The optimal value of “weightage factor” parameter needs to be calibrated by doing computational experiments.

CHAPTER 4: ANALYSIS OF DATA AND RESULTS

This chapter presents the details of computational experiments performed to test the performance of the post-processing technique developed in this study.

4.1 Computational Environment and the Test Networks

The computational experiments have been carried out to test the effectiveness of the post-processing techniques described in the previous chapter. For this purpose Sioux Falls, Anaheim and Borman Corridor networks have been taken as the test networks. Sioux Falls network consists of 24 nodes, 76 links and 552 O-D pairs with non-zero demand. The Anaheim network consists of 416 nodes, 914 links, and 1406 O-D pairs with non-zero demand. The Borman Corridor network in northwest Indiana consists of 197 nodes, 460 links and 1681 O-D pairs with non-zero demand. All the algorithms were coded in MATLAB and computational experiments were carried out using Dell precision workstation with Intel Xeon processors (2.67 GHz) with 24 GB RAM and 64-bit Windows 7 operating system. The convergence level for all the computational experiments were measured in terms of normalized gap ($Ngap$), and was assumed to be $Ngap$ equal to 10^{-6} .

4.2 Relative Performance of the Algorithms

As mentioned earlier, SMPA lies in the core of the post-processing module, which can be used as both, an independent traffic assignment algorithm, or as a post-processing technique using warm start. Hence it becomes imperative to test its relative performance with other potential algorithms in practice. This section presents the results of the computational experiment carried out to benchmark the relative performance of SMPA with other traffic assignment algorithms including, the Frank-Wolfe (F-W) algorithm; the most famous algorithm in practice, the social pressure algorithm (labeled SocPr) which is an improved version of F-W algorithm developed by Kupiszewska and Vliet (1999) and recently developed gradient projection algorithm (labeled FGP) of Florian et al. (2009).

The relative performance of SMPA with respect to other traffic assignment algorithms has been presented in Figure 4.1 through Figure 4.3. As evident from these figures, SMPA even without warm start and O-D prioritization converges faster and outperforms other algorithms for all the three test networks. Here it is important to mention that although with the increasing number of iterations the value of UE objective function decreases monotonically, that is not the case with $Ngap$ but its overall trend is decreasing. Knowing this fact, it is evident from the Figure 4.1 through Figure 4.3 that convergence of SMPA is much smoother than the other algorithms.

4.3 Evaluating the Benefits of Warm Start Using Perturbation Assignment

This section presents the results of the computational experiments performed to test the benefits of warm start using the perturbation assignment. As mentioned in the chapter 3, perturbation assignment for the case 4

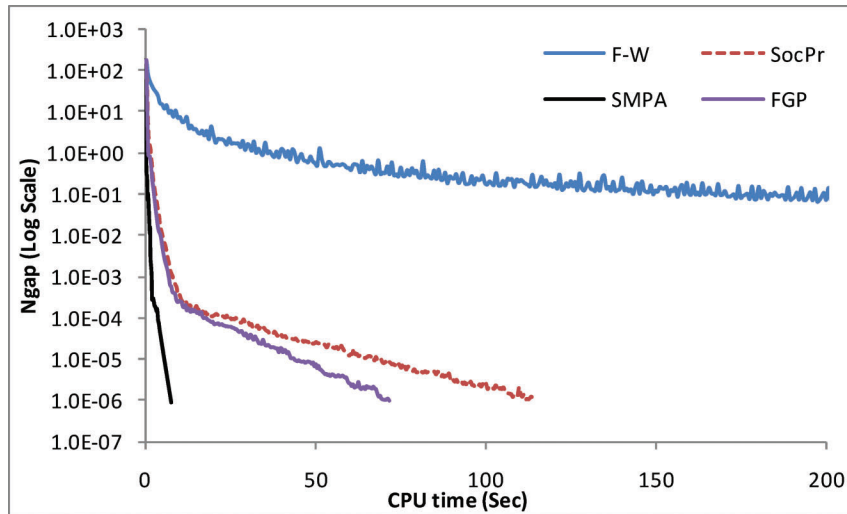


Figure 4.1 Convergence characteristics of the algorithms for the Sioux Falls network

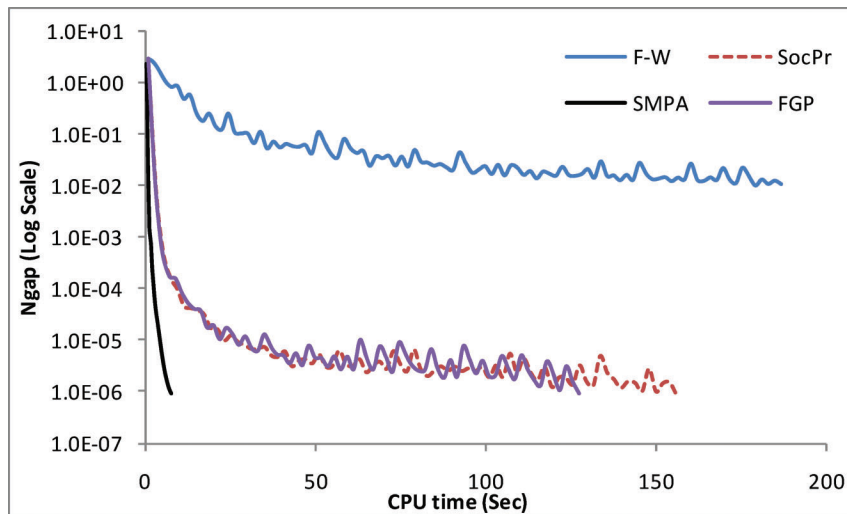


Figure 4.2 Convergence characteristics of the algorithms for the Anaheim network

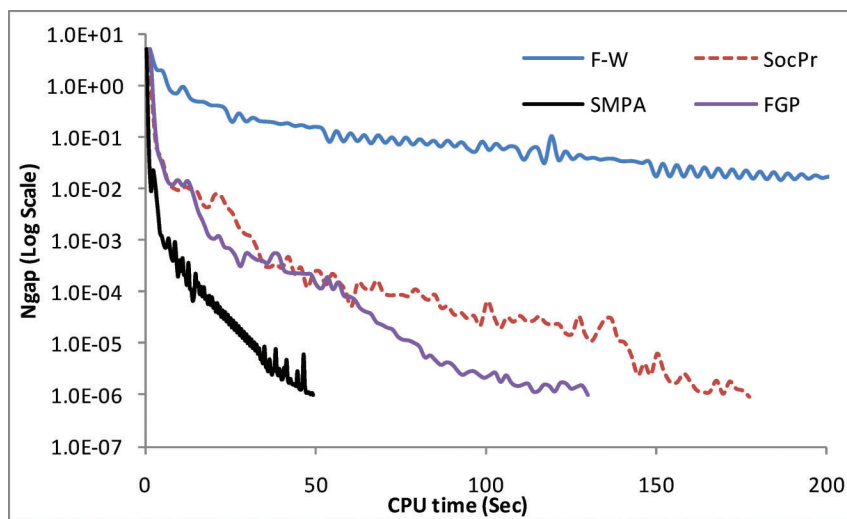


Figure 4.3 Convergence characteristics of the algorithms for the Borman network

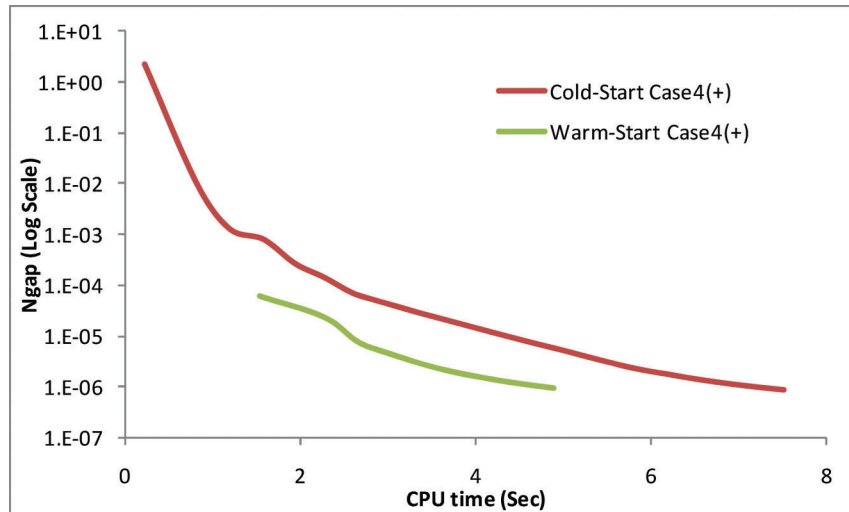


Figure 4.4 Cold-start and warm-start of SMPA for the Anaheim network with increased demand

involving the O-D demand change is most critical and has been selected for conducting the computational experiment. Under this case, where both the old and the new O-D demand for the O-D pair is non-zero, there can be two possible sub cases- the change in O-D demand can be positive meaning new O-D demand is higher than old demand (represented as case 4(+)) or change can be negative meaning that new O-D demand is less than the old O-D demand (represented as case 4(-)). Although implementation strategy for these two cases will be the same, the relative benefit of warm start under these two cases can be different and therefore investigated separately.

To test the benefit of warm start for case 4(+), 10 percent of the entries in O-D demand matrix were randomly selected and its values were increased by 10 percent. Similarly, to test the benefit of warm start for case 4(-), 10 percent of the entries in O-D demand matrix were randomly selected and its values were

decreased by 10 percent. Then computational experiments were conducted for Anaheim and Borman networks using SMPA with warm start. Figure 4.4 through 4.7 shows the results of the computational experiment. Although the relative benefits of warm start using SMPA varies from one case to another and with the networks, the results of the computational experiments indicate that savings in the computational time (CPU time) is significant.

4.4 Evaluating the Benefits of O-D Prioritization Techniques

As explained in chapter 3, the order in which O-D pairs are brought into equilibration process impacts the rate of convergence, this necessitates finding out the optimal O-D prioritization criteria. Computational experiments were performed to find out the best O-D prioritization criteria of SMPA for the Borman

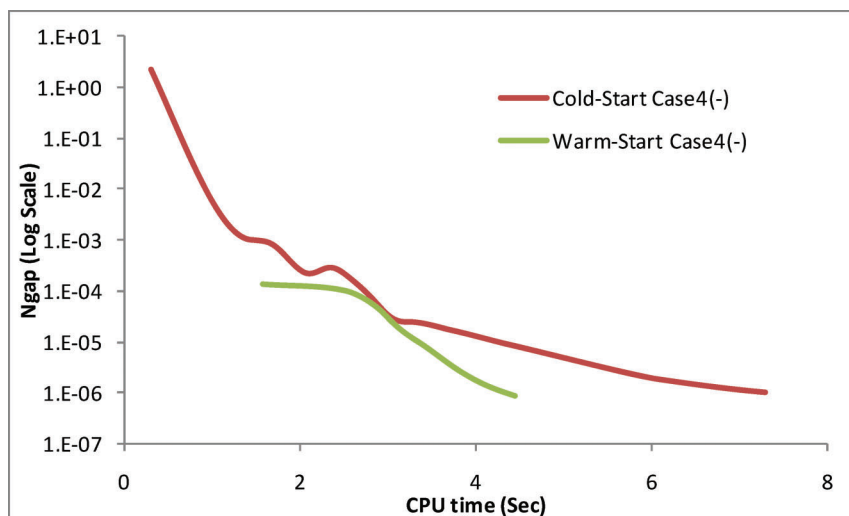


Figure 4.5 Cold-start and warm-start of SMPA for the Anaheim network with decreased demand

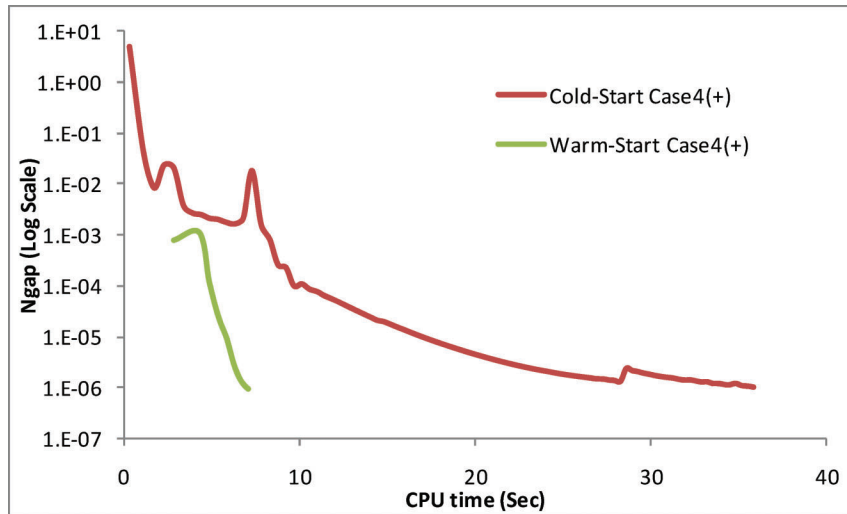


Figure 4.6 Cold-start and warm-start of SMPA for the Borman network with increased demand

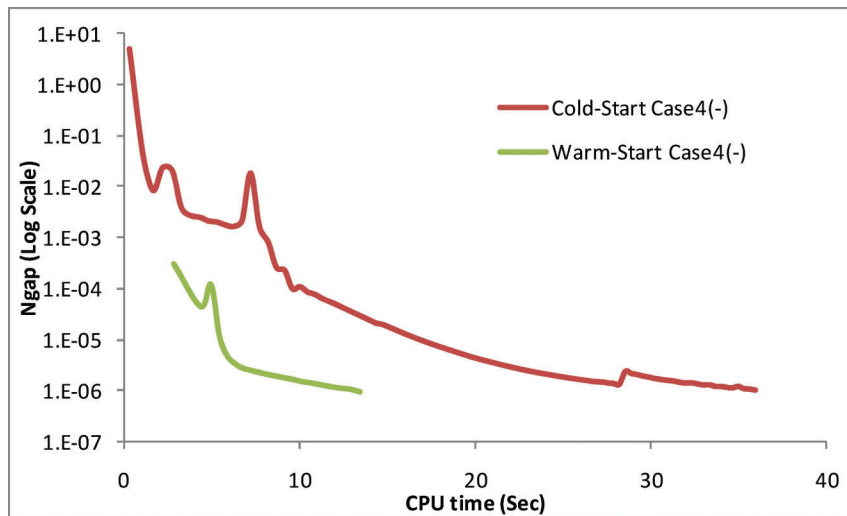


Figure 4.7 Cold-start and warm-start of SMPA for the Borman network with decreased demand

network. The SMPA was run for all the six criteria explained in chapter 3 along with the base case of no O-D prioritization, and the results obtained from these experiments are summarized in table 4.1. The two criterion having the weighted measure of O-D demand and free flow travel time ($ODPr=3$ and -3) requires the calibration of optimal *weightage factor* parameter. Computational experiments were performed to find the optimal *weightage factor* for these two cases ($ODPr=3$ and -3) and results of the same are summarized in table 4.2a and 4.2b.

Results of the computational experiments shows that optimal *weightage factor* for ascending and descending order of weighted measure of O-D demand and free flow travel time ($ODPr=3$ and -3) for the Borman network are 0.6 and 0.2 respectively. Results of the computational experiments also reveals that the ascending order of weighted measure of O-D demand and free flow travel time ($ODPr=3$) is the best O-D

prioritization criteria for this network. As clear from table 4.1, under this O-D prioritization criterion the convergence time for SMPA is 14.5 seconds, this is less than one third of the convergence time of the base case with no O-D prioritization ($ODPr=0$).

In order to test the combined benefit of warm start and O-D prioritization, computational experiments were performed with Borman network data. Figure 4.8 and 4.9 show the benefit of warm start for the O-D prioritized implementation of SMPA for this network.

4.5 Relative Benefits of Proposed Techniques on Link Flow Stability

Link flow stability is an important requirement for evaluating the transportation network improvement alternatives in practice. This requirement comes from fact that in practice, multiple alternatives needs to be evaluated, and due to time constraint assignment

TABLE 4.1
Deciding the optimal criteria of O-D prioritization for Borman network

Sr. No.	O-D Prioritization Criteria (ODPr)	Weightage Factor for O-D Prioritization	Number of iterations for Convergence	UE Objective Function at Convergence	Ngap at Convergence	CPU time (sec) for convergence
1	0	NA	123	408514.7	9.83E-07	49.4
2	1	NA	179	408514.7	9.03E-07	76.3
3	-1	NA	53	408514.7	9.52E-07	23.8
4	2	NA	50	408514.7	9.38E-07	21.2
5	-2	NA	52	408514.7	9.88E-07	22.4
6	3	0.6	33	408514.7	9.11E-07	14.5
7	-3	0.2	46	408514.7	8.34E-07	21.0

TABLE 4.2a
Deciding the optimal weightage factor of O-D prioritization for Borman network

Sr. No.	O-D Prioritization Criteria	Weightage Factor for O-D Prioritization	Number of iterations for Convergence	UE Objective Function at Convergence	Ngap at Convergence	CPU time (sec) for convergence
1	3	0.1	47	408514.7	9.76E-07	19.95
2	3	0.2	34	408514.7	9.81E-07	14.99
3	3	0.3	33	408514.7	9.8E-07	14.54
4	3	0.4	35	408514.7	9.2E-07	15.99
5	3	0.5	34	408514.7	9.64E-07	15.16
6	3	0.6	33	408514.7	9.11E-07	14.48
7	3	0.7	36	408514.7	9.43E-07	15.63
8	3	0.8	39	408514.7	9.34E-07	17.8
9	3	0.9	44	408514.7	9.92E-07	19.0
10	3	1	38	408514.7	9.3E-07	16.8

TABLE 4.2b
Deciding the optimal weightage factor of O-D prioritization for Borman network

Sr. No.	O-D Prioritization Criteria	Weightage Factor for O-D Prioritization	Number of iterations at Convergence	UE Objective Function at Convergence	Ngap at Convergence	CPU time (sec) for convergence
1	-3	0.1	107	408514.7	9.83E-07	43.96
2	-3	0.2	46	408514.7	8.34E-07	21.03
3	-3	0.3	200	Do not converge (DNC) in 200 iterations		
4	-3	0.4	114	408514.7	9.86E-07	47.52
5	-3	0.5	83	408514.7	9.91E-07	35.76
6	-3	0.6	93	408514.7	6.97E-07	37.53
7	-3	0.7	69	408514.7	9.94E-07	30.28
8	-3	0.8	200			DNC
9	-3	0.9	200			DNC
10	-3	1	200			DNC

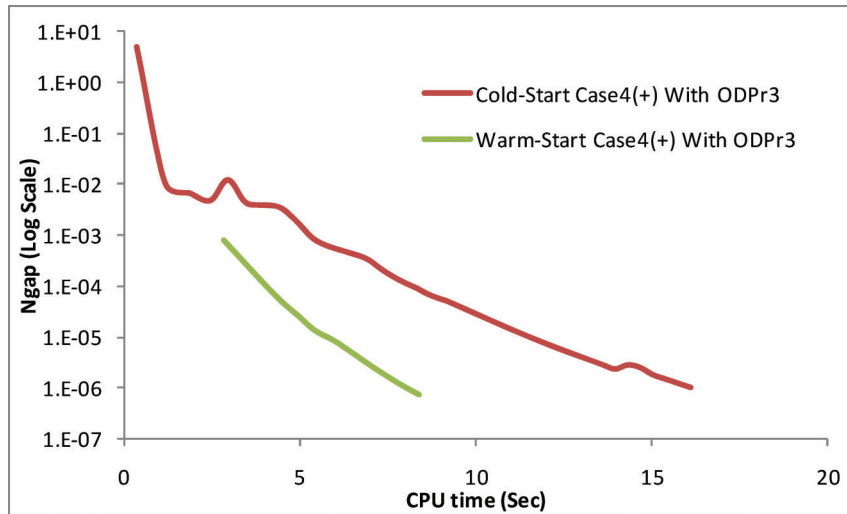


Figure 4.8 Combined benefits of warm-start and O-D prioritization using SMPA for the Borman network with increased demand

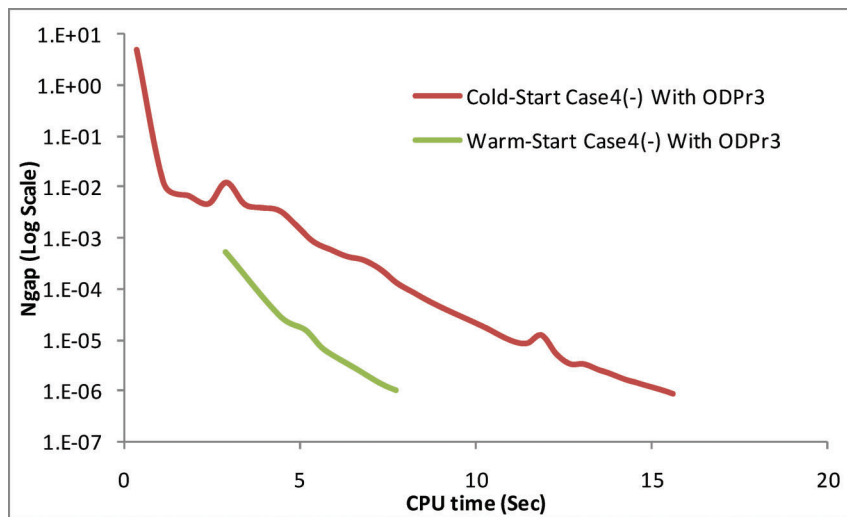


Figure 4.9 Combined benefits of warm-start and O-D prioritization using SMPA for the Borman network with decreased demand

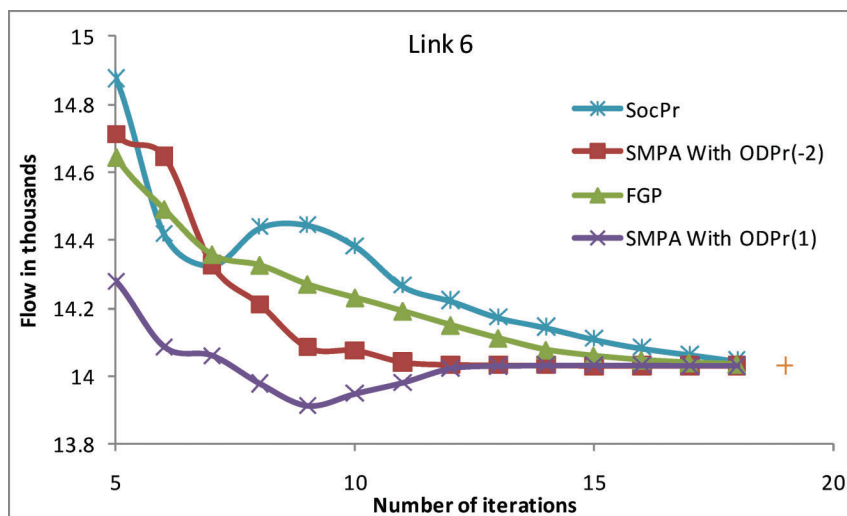


Figure 4.10 Link flow stability (Link 6) for Sioux Falls network

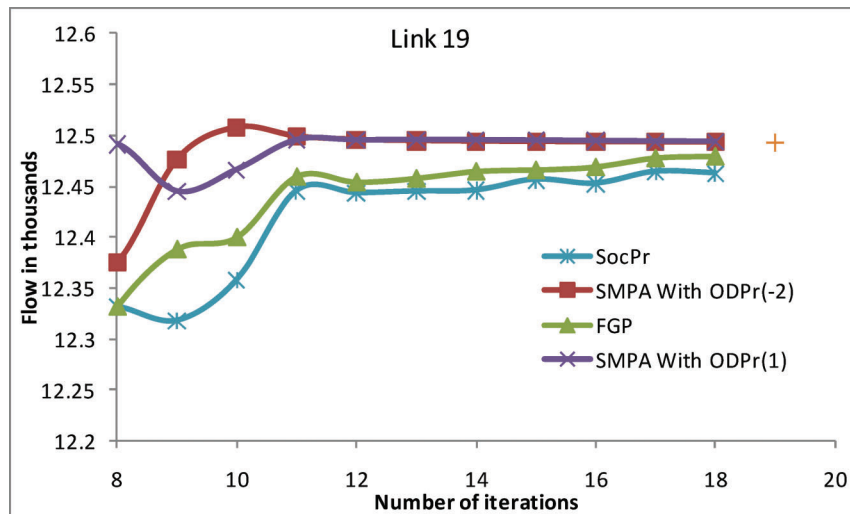


Figure 4.11 Link flow stability (Link 19) for Sioux Falls network

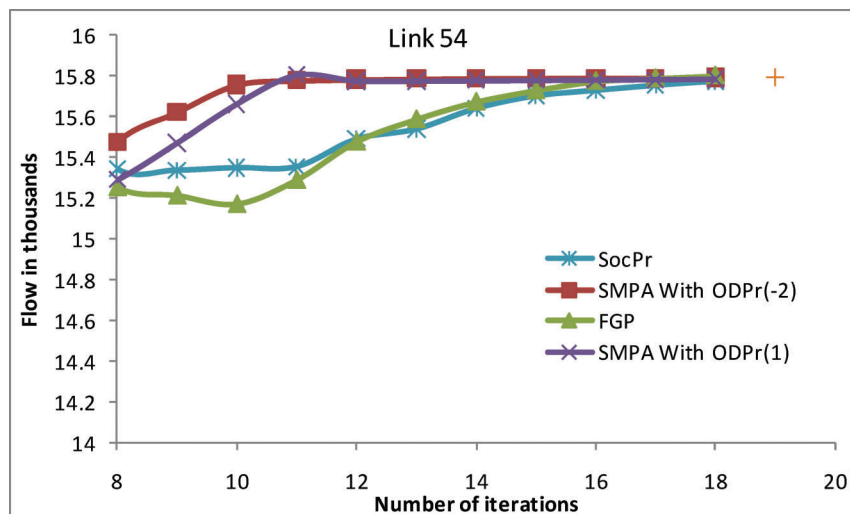


Figure 4.12 Link flow stability (Link 54) for Sioux Falls network

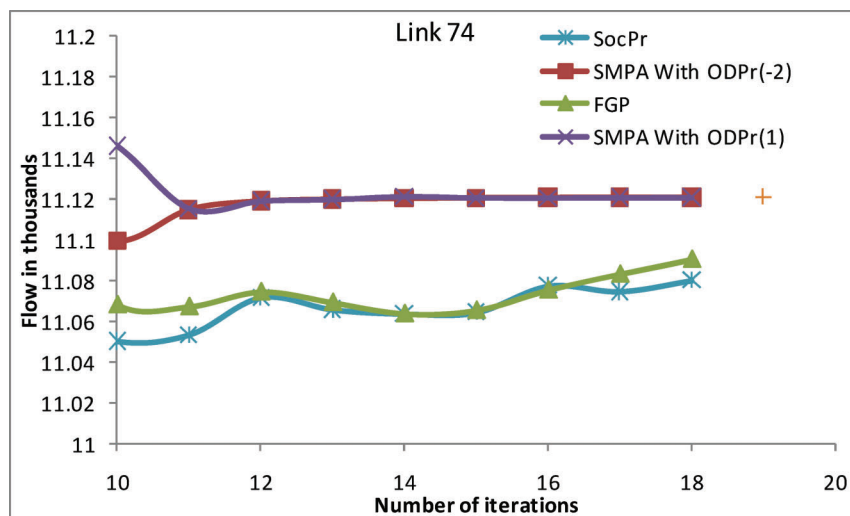


Figure 4.13 Link flow stability (Link 74) for Sioux Falls network

algorithms are cut off at some point even if it has not reached the convergence. Hence, usefulness of the algorithm requires lesser noise and oscillations in the link flows. Figures 4.10 through Figure 4.13 show relative performance of O-D prioritized implementation of SMPA with respect to other algorithms in terms of noise and stability of link flows for some randomly selected links of Sioux Falls network. The center of (+) sign on these plots represents the link flows at the converged solution. These plots indicate that link flows obtained by O-D prioritized SMPA is more stable and reaches near the equilibrium value at earlier iteration compared to other algorithms.

CHAPTER 5: EXPECTED BENEFITS, DELIVERABLES AND IMPLEMENTATION

5.1 Expected Benefits

The improved planning framework with post-processing technique developed in this study will provide a better solution with lesser noise and higher level of convergence compared to the conventional four-step planning process. In addition, solution obtained by adopting this methodology will have following important characteristics which will increase the reliability of the assignment algorithm:

Stability: Network flows obtained by this method will be stable and will have lesser noise in the solution

Consistency: Solution obtained in terms of link flows will be consistent which is particularly important while evaluating the multiple alternatives for network improvement.

Uniqueness: Uniqueness here means that the final solution obtained in terms of the link flows for a given network topology and link characteristics will be the same irrespective of the starting point by warm start. In addition, the same solution will be obtained for every run of the post-processing module if the input does not change.

5.2 Deliverables

The deliverables of this project include:

1. Executable code for SMPA integrated with O-D prioritization and perturbation assignment with following properties:
 - a. Generalized code which can be used for any network
 - b. Can be used on any computer with sufficient memory
 - c. Can be used for assignment step as well as for post-processing
2. Supporting files
 - a. MATLAB Runtime environment (MRE)
 - b. Sample data files
3. Guidelines showing the required format for input files and implementation steps

5.3 Implementation and Systematic Guidelines

This section provides the implementation methodology and systematic guidelines which will direct through the step by step process for implementing the post-processing techniques. This section also provides guidelines for data formatting required in implementing the executable code of the post-processing module.

5.3.1 Implementation Methodology

The philosophy of implementation of post-processing techniques with the other steps of sequential planning process has been shown by the implementation flow chart in Figure 5.1.

First four steps of the sequential planning process will be carried out in TransCAD and the solution in terms of paths, path flows along with the O-D demand matrix will be saved as output files. The post-processing module will use the paths, path flows and the O-D demand matrix and provide a more precise UE solution having lesser noise. This solution will be in terms of path flows, link flows and link costs. The link costs obtained from the post-processing module will be utilized by the TransCAD for updating the O-D demand matrix. This new demand matrix after the modal split process in the TransCAD will provide the auto trip matrix which will be fed back to post-processing module. This process will continue till average of absolute percentage change in the link flows becomes less than a threshold value as explained in section 3. Hence it is clear that it will involve transfer of data between post-processing module and TransCAD. Post-processing module will accept data only in a particular format. This can be done manually or by developing preprocessing module which will convert data obtained from TransCAD to required format.

5.3.2 Systematic Guidelines

Following sequence of steps should be followed for implementing the post-processing module along with the four-step planning process-

Calibration step: In this step best O-D prioritization criterion for the given network is found by using the flow chart shown in Figure 3.7. The paths, path flows and O-D demand obtained from TransCAD is used to run the post-processing module based on all the six criteria mentioned above. For this purpose the input parameter in the required file (input_parameters.txt) needs to be changed as explained in the next section. Then the best O-D prioritization criterion is decided by comparing the results based on the flow chart (Figure 3.7). This calibration step is required to be carried out only once for a given network. This criterion will hold good for this network even if network topology changes slightly.

Main Steps: Once the best O-D prioritization criterion is established for a given network, the following sequence of the steps is followed:

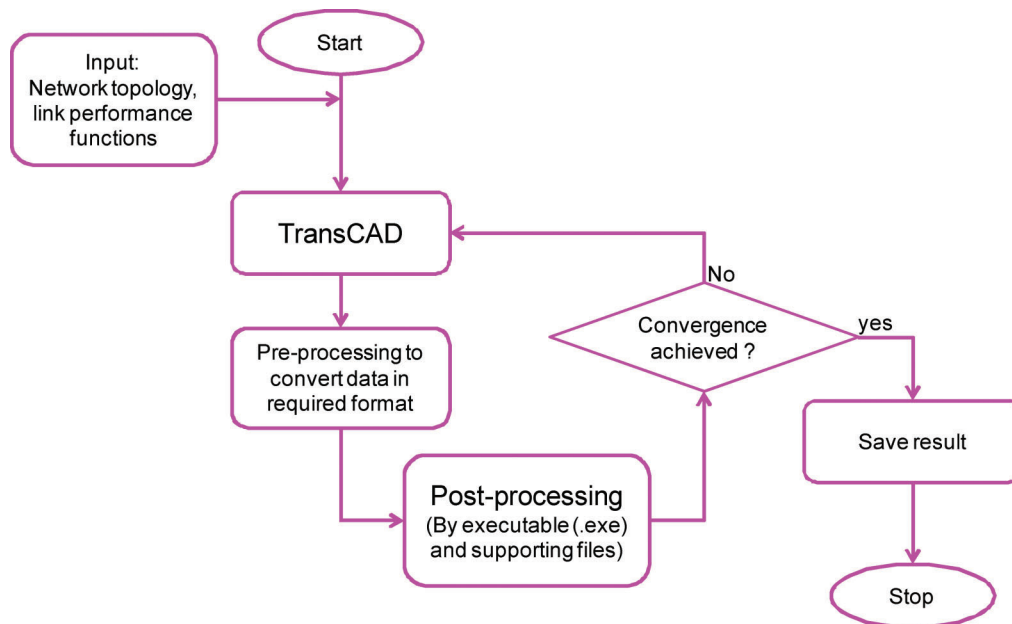


Figure 5.1 Implementation methodology

Step 1: Run the first four steps of planning process in TransCAD to obtain the trip matrix and approximate UE solution in terms of path flows.

Step 2: Do the preprocessing to convert the data in the necessary format for the post-processing module

Step 3: Run the post-processing module by double clicking the executable code (or alternatively by calling it from the TransCAD). This will run the post-processing step and save the paths, path flows, link flows and link costs as separate files.

Step 4: Use the link costs obtained from the post-processing module and generate the new trip matrix by trip generation module and then find the auto share by modal split (mode choice) module in TransCAD and rename the output file (linkout.txt) of post-processing module as linkout_old.txt.

Step 5: Do the preprocessing to convert the new trip matrix in the necessary format for post-processing module and run the post-processing module by double clicking the executable file or calling it from the TransCAD. This will run the post-processing step and save the paths, path flows, link flows and link costs as two separate files.

Step 6: Run the link flow comparison module to find the average absolute percentage change in link flows. Check the termination criteria. If a termination criterion is satisfied stop, else go to step 4.

Guidelines for data formatting:

As mentioned earlier, the executable code for the post-processing module will work only when the input files are provided in proper format. These input files should be saved with exactly the same name and file type as mentioned further. The executable code will also write the output of post-processing process in the output files. But they are not the critical part because if

the output files are not present they will be created by the code, and data will be saved in proper format. The code overwrites the new data if the output files carry previously saved data in it. The input files required for the post-processing module are as below:

1. input.xls
2. link.txt
3. origin_nums.txt
4. destin_nums.txt
5. demand.txt

The output files of the post-processing module and their file extensions are as below:

1. linkout.txt
2. pathdata.mat

The details of all the above mentioned input and output files along with the required format for preparing these files are presented below.

input.xls

This file contains the parameters which controls the number of iterations and provides the other parameters required for running the post-processing module. This file should be saved as input.xls and should have three sheets named respectively as “parameters”, “statistics” and “linkout”. The input is required to be provided only in column B of the sheet named “parameters”. The details of this sheet(named “parameters”) are shown in Figure 5.2.

Details of sheet named “parameter”s in input.xls

The first and second elements of the column B provides input about the number of origins and number of destinations respectively. These entries should be positive integers. Although they are automatically calculated from the demand data files these two elements of the column B should not be left blank.

	A	B	C
1	Number of origin nodes	38	
2	Number of destination nodes	38	
3	Maximum number of outer (main) iterations	200	
4	Termination threshold (Normalized Gap)	1.00E-05	
5	Scaling factor for move direction	1.6	
6	Maximum number of inner iterations	9	
7	Level of Demand (fraction ,positive)	1	
8	Weight factor for O-D prioritization	0.6	
9	First through node	39	
10	O-D prioritization criteria	0	
11	Warm start (0 if no, 1 if yes)	0	
12			

Figure 5.2 Details of input.xls

The third element of column B in this sheet specifies the maximum number of outer iterations of SMPA. This acts as a safety against running the algorithm for very long time. It should be a positive integer greater than 1 (preferably 200).

The fourth element of column B in this sheet decides the stopping criteria for the SMPA and provides the threshold limit at which convergence is considered to be achieved. It should be a negative power of base 10. Scientific format is preferred for this element as shown in Figure 4.1.

The fifth element provides scaling factor for the move direction to the SMPA. It can be any positive number (integer or fraction).

The sixth element restricts the number of inner loop of SMPA. It can be any positive integer preferably 9.

The seventh element provides the level of demand. Post-processing module multiplies the trip matrix with this number. It can be any positive number (integer or fraction). In most cases it is likely to have a value 1 except when sensitivity analysis with respect to demand need to be carried out.

The eighth element provides weightage factor for O-D prioritization scheme. This gives relative importance of free flow travel time with respect to O-D demand in O-D prioritization and is used only when the criterion for O-D prioritization is 3 or -3. Even if the O-D prioritization is not used it should not be left blank but should be provided as zero.

The ninth element gives the node number of first real node (physical node) or through node. Here it is assumed that all centroid connector numbers are less than the first through node number.

The tenth element provides the O-D prioritization criteria for the post-processing module. The input values corresponding to the seven O-D prioritization criteria are presented in table 5.1.

Eleventh element of the column B of this sheet (named “parameters”) of the input.xls file controls the initialization step of the module. It can have only two values 0 or 1. If this entry is 0 (zero), the module invokes the cold start process of initialization using all-or-nothing assignment. And if the entry in this cell is 1 then module invokes initialization with warm start and uses the path flows saved from previous runs. The entry

TABLE 5.1
O-D prioritization criteria and their parametric values

SN	O-D prioritization criteria	Value
1	No O-D prioritization	0
2	Ascending order of O-D demand	1
3	Descending order of O-D demand	-1
4	Ascending order of free flow travel time	2
5	Descending order of free flow travel time	-2
6	Ascending order of weighted measure of O-D demand and free flow travel time	3
7	Descending order of weighted measure of O-D demand and free flow travel time	-3

in this cell should be 1 only when result of the previous run is available.

Details of sheet named “statistics” in input.xls

This sheet saves the statistics of the run of SMPA module in four columns. The first column gives the iteration number, the second, third and fourth column gives respectively the values of UE objective function, normalized gap (Ngap) and CPU time at different iterations. This sheet is overwritten every time the module is run. But care should be taken to use this sheet, as the number of rows which are overwritten is equal to the number of iterations required to achieve convergence. Hence, if this sheet is of interest then after every run, the result saved in this sheet should be copied and then the entries in this sheet should be cleared.

Details of sheet named “linkout” in input.xls

This sheet provides the output of link. Same can also be obtained from linkout.txt

link.txt

This file provides the link properties to the post-processing module. In this file, properties of a link are represented by a row of data. A link is represented by start node and end node. If a link is directional (one way) then its direction value (last column) is 1. If a link is bidirectional then it can be represented by two ways. First method: representing it by two rows with swapped start and end node and the value 1 in last column (representing direction). Second method: representing it by single row and by putting the nodes which it connects (in column 1 and 2 without bothering about the start node and end node) and putting the value 0 in the last column which represents direction. Elements of the rows in this file are separated by tab character.

origin_nums.txt

This file lists the origin numbers as a column vector. The origins numbers are listed in ascending order. These numbers may not be continuous but should be positive integers. Fractional numbers cannot be used for numbering the origins.

destin_nums.txt

This file lists the destination numbers as a column vector. The destination numbers are listed in ascending order. The entries of this file are same as the origin numbers but just the different file name.

link - Notepad

File	Edit	Format	View	Help					
231560	42007	1000	1.09	2.617	0.15	4	25	0	0
202123	49001	1000	2.69	6.446	0.15	4	25	0	0
202191	49001	1000	2.34	5.626	0.15	4	25	0	0
204819	49016	1000	1.11	2.667	0.15	4	25	0	0
204822	49016	1000	0.95	2.285	0.15	4	25	0	0
210453	49016	1000	1.10	2.634	0.15	4	25	0	0
210456	49022	1000	1.04	2.488	0.15	4	25	0	0
210468	49022	1000	0.58	1.387	0.15	4	25	0	0
215275	49040	1000	0.91	2.178	0.15	4	25	0	0
202655	49042	1000	0.79	1.888	0.15	4	25	0	0
210511	49093	1000	0.42	1.004	0.15	4	25	0	0
232336	49093	1000	0.44	1.045	0.15	4	25	0	0
231968	67026	1000	2.91	6.979	0.15	4	25	0	0
207203	82001	1000	0.68	1.620	0.15	4	25	0	0
226939	82001	1000	0.86	2.056	0.15	4	25	0	0
226815	82004	1000	0.92	2.213	0.15	4	25	0	0
231444	82004	1000	0.80	1.927	0.15	4	25	0	0
207690	82006	1000	0.60	1.432	0.15	4	25	0	0
231284	82011	1000	0.58	1.391	0.15	4	25	0	0

Startnode

Endnode

Capacity

Length

Free flow travel time

Beta

Power

Posted speed

Toll

Direction

Figure 5.3 Details of link.txt

```

File Edit Format View Help
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019

```

Figure 5.4 Details of origin_nums.txt

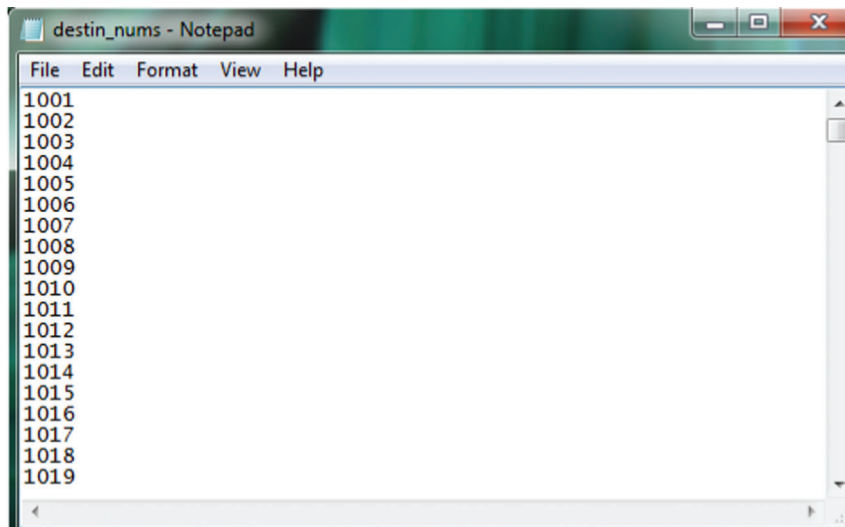


Figure 5.5 Details of destin_nums.txt

A screenshot of a Notepad window titled "linkout - Notepad" showing a table of data. Below the table are seven vertical labels with arrows pointing to the corresponding columns: From Node, End node, Capacity, Free flow travel time, Travel time, Volume, and Slope of cost function.

From Node	End node	Capacity	Free flow travel time	Travel time	Volume	Slope of cost function
168	409	1800	0.5	0.64805	2133.6	0.00027756
408	409	5400	0.5	0.5002	1219.8	6.40E-07
410	409	5400	0.22008	0.22008	650.4	4.27E-08
396	410	5400	2	2	2	1.13E-14
409	410	5400	0.22008	0.22008	594.26	3.26E-08
411	410	5400	2	2.0001	722.1	5.31E-07
8	411	5400	1	1	722.1	2.66E-07
410	411	5400	2	2	37	7.15E-11
21	412	5400	1	1.0007	1396.5	1.92E-06
402	412	5400	2	2	570.93	2.63E-07
21	413	5400	1	1.0004	1245.3	1.36E-06
404	413	5400	2	2.0017	1489	4.66E-06
22	414	5400	1	1	619.8	1.68E-07
405	414	5400	2	2	509.4	1.87E-07
22	415	5400	1	1.0001	904.6	5.22E-07
406	415	5400	2	2.0003	934.2	1.15E-06
23	416	5400	1	1.0009	1522.5	2.49E-06
407	416	5400	2	2	387.9	8.24E-08

Figure 5.6 Details of linkout.txt

demand.txt

This file also acts as input to the post-processing module. It is a square matrix, where each row represents the trip demand from an origin zone to all destinations. Trip demands are entered in ascending order of destination numbers in each row and ascending order of origin numbers along each column. Elements of rows are separated by tab character.

linkout.txt

This is the output file of post-processing module. It gives the link travel time which will be used in the trip distribution step by feedback loop. It has a structure similar to the link.txt but some columns are different.

pathdata.mat

This is the output file of the post-processing module. It contains the path and path flows for all the O-D pairs. It is used by the post-processing module for warm start if feedback loop is used more than once.

CHAPTER 6: RECOMMENDATIONS

6.1 Recommendations for Efficient Utilization of the Post-Processing Techniques

The recommendations for efficient utilization of the post-processing techniques developed in this study include:

1. **Optimal calibration of the O-D prioritization criterion:** The O-D prioritization criterion for the given network can be obtained by using the flow chart shown in Figure 3.7. The optimal O-D prioritization criterion will differ from one network to other. The optimal O-D prioritization criteria may also change if the O-D demand changes significantly resulting in significant change in the level of congestion in the network.
2. **Proper order of evaluation of network improvement alternatives:** Warm start is not only useful for faster execution of traffic assignment for a slightly different demand resulting due to feedback loop, but also for slightly different network, for example alternatives involving capacity expansion of few links or addition of few links in the network. For using the warm start by perturbation assignment technique for evaluating the alternative involving addition or deletion of links following criteria should be followed-
 - a. While evaluating the two alternatives, if the set of links A_1 for an alternative network is the subset of the other network A_2 , then first run the travel demand model with the post-processing module for smaller network represented by A_1 and save the result in terms of path and path flows for this network. Then use the saved result for warm starting the assignment for larger network using the post-processing module followed by feedback loop. If the additional link in A_2 which is not in A_1 will improve the travel time (cost) for any O-D pair it will get included by SMPA automatically.
 - b. If the set of links A_1 for an alternative is not a subset of other network A_2 , start with cold start for evaluating both alternatives
 - c. If there are multiple alternatives and set of links A_1 of an alternative is subset of all other alternative.

Then, then first run the travel demand model with the post-processing module for smallest network represented by A_1 and save the result in terms of path and path flows for this network. Then use this saved result for warm starting the assignment for all other alternatives using the post-processing module followed by feedback loop.

6.2 Recommendations for State-Wide Planning

In this study the developed post-processing techniques were tried to run on the Indiana State-Wide Model. The initial results suggested that the executable code of the developed post-processing module in present form is not an efficient implementation strategy for such a large size network. The first reason for this is due to the fact that the code was developed in MATLAB platform which is good for small and medium size city networks but not good for the network of very large size such as Indiana State-Wide network. For the present study, the aim was to test the developed concepts for its potential benefits and MATLAB provided a scientific computing platform for executing these concepts. But for its implementation on a state-wide network, these concepts need to be efficiently coded in other computing language like C or C++. The second reason comes from some inherent characteristics of the state-wide network. By a closer look into the data sets of the network it was found that many O-D pairs of the network have zero and very small demand. O-D pairs with zero demand get skipped by the executable code but difficulty comes when demand is very small. Even though these O-D demands are insignificant, it cannot be skipped due to fact that collectively they can be significant especially where there are millions of O-D pairs (e.g. Indiana State-Wide network).

For the state-wide planning, a better implementation strategy will be to use the two stage modeling approach. The state-wide network should be divided into clusters. Then at the first stage, the travel demand modeling should be done for each cluster separately. As the travel demand modeling of each cluster will be independent from the other, it can be done in parallel with the multi-core processors. Then at the second level, each cluster will act functionally as a single functional zone. Travel demand modeling should be done by taking these bigger functional zones as possible origins and destinations. The level of network details will differ from first stage to second stage. While the first stage will have minor streets as well as higher level links, the second stage will have only those streets and highways (links) which are required to connect the clusters (bigger functional zones). The links which will appear in the second stage will carry background traffic from the first stage.

As these are just the preliminary thoughts, recommendations for a good criteria for clustering are not eminent but will need a detailed research. But some suggestions for demarcating the boundaries of the cluster are as below:

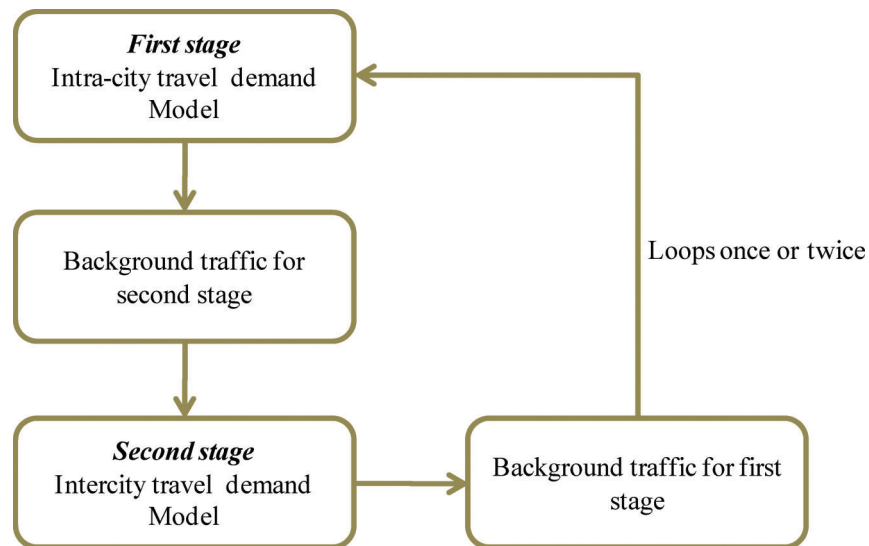


Figure 6.1 Recommended state-wide planning model

1. City boundaries can be used as demarcating the boundaries for the clusters
2. Twin cities having commuting trips (eg. Lafayette and West Lafayette, in the state of Indiana) can be kept in a single cluster

Above clustering approach means that first stage will model the intra-city demand considering two cities in the close proximity as one city, and the second stage will model the intercity travel demand. The background traffic resulting from the first stage will impact the travel decisions on the second stage. Here, it is arguable that the traffic flow resulting from the intercity travel may also impact the intra-city travel. This problem can be tackled by looping back from the second stage to the first stage as shown in the flow chart in Figure 6.1.

CHAPTER 7: CONCLUSIONS

This chapter summarizes the research, highlights its contributions, outlines the limitations of the present study associated with the methodology developed in this project and proposes directions for future research.

7.1 Research Summary

In this study, a post-processing technique was developed to solve the problem of stability, consistency and convergence of traffic assignment algorithms in practice. The developed post-processing module is implemented through the feedback loop in the four-step transportation planning process. In addition to improving the stability and convergence properties of the assignment solution, this feedback loop is also useful in solving the problem of interdependency of second and fourth steps of the planning process.

The post-processing module consists of three building blocks namely, slope-based multi-path algorithm or SMPA, perturbation assignment and O-D prioritization technique. SMPA is the most important part of the

module and it can be used as both post-processing algorithm or as an independent static traffic assignment algorithm. It uses a sequential O-D equilibration scheme where O-D pairs are brought into equilibration process one at a time in sequential order. Perturbation assignment helps to utilize the results of previous runs of the assignment algorithm and provides warm start to SMPA. O-D prioritization technique decides the order in which O-D pairs are brought into equilibration process in sequential assignment algorithm and catalyzes the rate of convergence of SMPA.

Computational experiments were performed to test the effectiveness of the post-processing techniques. Results of the computational experiments reveal that SMPA has a superior rate of convergence compared to state of practice algorithms. Results of computational experiments further reveals that warm start using perturbation assignment and O-D prioritization has significant benefit over the base case of cold start and non-prioritized implementation of SMPA. These three techniques will improve the convergence characteristics of the assignment process and provide a more stable solution having lesser noise and thereby increasing the reliability of planning process. Efficient use of solution of previous run the assignment process by perturbation assignment will also be helpful in comparing the transportation network improvement alternatives which differs slightly, for example, alternative involving the capacity expansions of the links.

7.2 Contributions

In this study three practical issues namely, stability, consistency and convergence related to the traffic assignment algorithm were identified and a post-processing module was developed to deal with these problems. Potential traffic assignment algorithms were studied from the literature and its merits and demerits were investigated. Taking insights from potential

algorithms in literature, a new traffic assignment algorithm labeled slope-based multi-path algorithm or SMPA was developed which incorporates the merits of those algorithms, but its flow update mechanism is new and differs from them. To investigate the relative merits components of post-processing module, computational experiments were performed using three networks and their results were studied.

From the methodological and practical implementation point of view, the contributions of this study are:

1. ***A new solution algorithm for UETAP (section 3.2):*** This study defines the static user equilibrium traffic assignment problem (UETAP) by decomposing the objective function of Beckmann's transformation into three parts and formulates a new solution algorithm labelled slope-based multi-path algorithm or SMPA. It has a better convergence characteristics compared to other potential algorithms in practice.
2. ***Development of a hybrid approach for UETAP solution algorithm (section 3.2.3 and 3.2.4):*** A hybrid approach was developed by combining the merits of simultaneous and sequential approach to foster fast implementation of UETAP algorithms for large size networks and was executed in SMPA. In this approach shortest paths are generated and set of paths are updated for all the O-D pairs simultaneously and then paths for each O-D pair are equilibrated and flows are updated based on the sequential approach.
3. ***Detailed implementation procedure for Perturbation assignment (section 3.3):*** The technique of perturbation assignment was studied for testing the potential of utilizing the information from previous run of the assignment algorithm for slightly different input. A detailed implementation procedure was developed to facilitate its seamless implementation.
4. ***Development of O-D prioritization technique (section 3.4):*** For the assignment algorithms using sequential equilibration techniques, the order in which the O-D pairs are brought into flow update process can have significant impact on the rate of convergence and the solution stability. In this study an implementation methodology for O-D prioritization technique was developed and six criteria for O-D prioritization were conceived and tested for real-size network. Computation experiments were performed for investigating the benefit of this technique in improving the rate of convergence and stability of link flows.

7.3 Limitations of the Present Study and the Post-processing Technique

From the methodological point of view, one of the restrictive limitations of the present study is that it works only with the path-based approach and need to store paths and path flows. This leads to higher memory requirements compared to link based approach such as F-W algorithm which are popular in practice. But due to fast evolving computational capabilities of today, this requirement is no more problematic. From the implementation point of view, limitations of the present study are:

1. Present study used the small and medium size networks for the computational experiments to test the benefits of the post-processing techniques
2. Post-processing techniques was tested on MATLAB platform which is computationally slower than other programming language like C or C++. Hence, performance of the post-processing module can be further improved if those concepts are coded more efficiently.
3. Most of the execution involved coding which is in serial fashion and hence could not be benefitted from multiple cores of computers. More efficient use of parallel processing can further improve the performance.

7.4 Future Research Directions

As mentioned in the previous section, the performance of the algorithms can be improved by efficient use of parallel processing. In this project all the data were processed with the central processing units (CPU) of the computer, but the same can be done by using the graphics processing units (GPU) more efficiently. GPU computing harnesses the capacity of GPU to do general purpose scientific and engineering computing by new massively parallel architecture called "CUDA". In heterogeneous coding requirement where part of an application is in serial and other parts can be performed in parallel, an efficient model will be to combine the power of both CPU and GPU. This will mean to use a CPU and GPU together in a co-processing computing model. The sequential or serial part of the application will run on the CPU and the computationally-intensive part is accelerated by the GPU by running in parallel. This will be particularly helpful for implementing the computationally intensive concepts for very large networks such as state wide network and mega regions.

REFERENCES

- Bar-Gera, H., (2002) Origin-Based Algorithm for the Traffic Assignment Problem. *Transportation Science*, 36(4), 398–417.
- Bar-Gera, H., (1999) *Origin-Based Algorithms for Transportation Network Modeling*. PhD Dissertation, University of Illinois at Chicago. Technical Report No. 103.
- Beckmann, M., McGuire, C. B., and Winsten, C. B., (1956) *Studies in the Economics of Transportation*. Yale University Press, New Haven.
- Bertsekas, D. P., (1976) On the Goldstein-Levitin-Poljak Gradient Projection Method. *IEEE Transactions on Automatic Control*, 21(2), 174–184.
- Bertsekas, D. P., and Gallager, R. G., (1987) *Data Networks*. Englewood Cliffs, Prentice-Hall, NJ.
- Bruynooghe, M., Gilbert, A., and Sakarovitch, M., (1969) Une Méthode d'affectation du Traffic. In: Leutzbach W, Baron P (eds). *Proceedings of the 4th International Symposium on the Theory of Road Traffic*. Bundesminister für Verkehr, Abt. Strassenbau, Bonn, Germany.
- Boyce, D., Ralevic-Dekic, B., and Bar-Gera, H., (2004) Convergence of Traffic Assignments: How Much is Enough? *Journal of Transportation Engineering, ASCE*, 130(1), 49–55.

- Chen, A., Lee, D. H., and Jayakrishnan, R., (2002) Computational study of state-of-the-art path based traffic assignment algorithms. *Mathematics and Computers in Simulation*, 59, 509–518.
- Dafermos, S., (1968) *Traffic Assignment and Resource Allocation in Transportation Networks*. PhD Dissertation, Johns Hopkins University, Baltimore, MD.
- Dafermos, S., and Sparrow, R., (1969) The Traffic Assignment Problem for a General Network. *Journal of Research of the National Bureau of Standards*, 73B, 91–118.
- Dial, R.B., (2006) A Path-Based User-Equilibrium Traffic Assignment Algorithm That Obviates Path Storage and Enumeration. *Transportation Research Part B*, 40(10), 917–936.
- Dow, P. and Vliet, V. D., (1979) Capacity restrained road assignment. *Traffic Engineering and Control*, 20, 261–273.
- Florian, M. and Nguyen, S., (1976) An Application and Validation of Equilibrium Trip Assignment Methods. *Transportation Science*, 10, 374–389.
- Florian, M., Constantin, I., and Florian, D., (2009) A New Look at the Projected Gradient Method for Equilibrium Assignment. CD-ROM. Transportation Research Board of the National Academics, Washington, D.C.
- Florian, M., Guelat, J., and Spiess, H., (1987) An Efficient Implementation of the “PARTAN” Variant of the Linear Approximation Method for the Network Equilibrium Problem. *Networks*, 17, 319–339.
- Frank, M., and Wolfe, P., (1956) An Algorithm for Quadratic Programming. *Naval Research Logistics Quarterly*, 3, 95–110.
- Fukushima, M. A., (1984) Modified Frank-Wolfe Algorithm for Solving the Traffic Assignment Problem. *Transportation Research Part B*, 18(2), 169–177.
- Gallager, R. G., (1977) A Minimum Delay Routing Algorithm Using Distributed Computation. *IEEE Transactions on Communications*, 25(1), 73–85.
- Goldfarb, D. and Spielberg, F., (2005) Does the Emperor have Clothes? A Comparative Evaluation of Assignment Techniques, *10th TRB Transportation Planning Applications Conference*, Portland, OR, April 24–28.
- Jayakrishnan, R., Tsai, W. K., Prashker, J., and Rajadhyaksha, S., (1994) A Faster Path-Based Algorithm for Traffic Assignment. *Transportation Research Record: Journal of the Transportation Research Board*, No. 1443, Transportation Research Board of the National Academics, Washington, D.C., 75–83.
- Knight, F. H., (1924) Some Fallacies in the Interpretation of Social Cost. *Quarterly Journal of Economics*, 38, 582–606.
- Kumar, A. and Peeta, S., (2010) Slope-Based Multipath Flow Update Algorithm for the Static User Equilibrium Traffic Assignment Problem, *Transportation Research Record: Journal of the Transportation Research Board*, Transportation Research Board of the National Academics, Washington, D.C., 1(2196).
- Kupiszewska, D. and Vliet, D. V., (1998) Incremental Traffic Assignment: A Perturbation Approach. In: *Proceedings of the 3rd IMA International Conference on Mathematics in Transportation Planning and Control* (J. D. Griffiths, ed.), 155–165.
- Kupiszewska, D., and Vliet, D. V., (1999) 101 Uses for Path-Based Assignment. *Transport Planning Methods: Proc.*, Seminar C, PTRC Planning and Transport Summer Ann. Meeting, Univ. of Sussex, U.K.
- Larsson, T., and Patriksson, M., (1992) Simplicial Decomposition with Disaggregate Representation for the Traffic Assignment Problem. *Transportation Science*, 26(1), 4–17.
- LeBlanc, L. J., (1973) *Mathematical Programming Algorithms for Large Scale Network Equilibrium and Network Design Problems*. PhD Dissertation, Department of Industrial Engineering and Management Science, Northwestern University, USA.
- LeBlanc, L. J., Morlok, E. K., and Pierskalla, W. P., (1975) An Efficient Approach to Solving the Road Network Equilibrium Traffic Assignment Problem. *Transportation Research*, 9, 309–318.
- LeBlanc, L. J., Helgason, R. V., and Boyce, D. E., (1985) Improved Efficiency of the Frank-Wolfe Algorithm for Convex Network Problems. *Transportation Science*, 19(4), 445–462.
- Lee, D. H. and Nie, Y., (2001) Accelerating Strategies and Computational Studies of the Frank-Wolfe Algorithm for the Traffic Assignment Problem, *Transportation Research Record: Journal of the Transportation Research Board*, Transportation Research Board of the National Academics, Washington, D.C., 1771, 97–105.
- Luenberger, D. G., (1973) *Introduction to Linear and Nonlinear Programming*. Addison-Wesley, Reading, Massachusetts.
- Meyer, G., (1974) Accelerated Frank-Wolfe Algorithms. *SIAM Journal on Control*, 12, 655–663.
- Nguyen, S., (1974) An Algorithm for the Traffic Assignment Problem. *Transportation Science*, 8(3), 203–216.
- Rose, G., Daskin, M. S., and Koppelman, F. S., (1988) An Examination of Convergence Error in Equilibrium Traffic Assignment Models. *Transportation Research part B*, 22(4), 261–274.
- Rosen, B., (1960) The Gradient Projection Method for Nonlinear Programming, Part I. Linear Constraints. *Journal of the Society for Industrial and Applied Mathematics*, 8(1), 181–217.
- Sinha, K. C. and Labi, S. (2007) *Transportation Decision Making Principles of Evaluation and Programming*. John Wiley & Sons, Inc. New Jersey.
- VHB, (2006) Results of FY2006 Travel Forecasting Research, *Report to the Metropolitan Washington Council of Governments*, National Capital Region Transportation Planning Board, VanasseHangenBrustlin, Inc.
- VHB, (2007) Determination of the State of the Practice in Metropolitan Area Traffic Forecasting: Findings of the Surveys of Metropolitan Planning Organizations, *Report to the Transportation Research Board Committee B0090*, VanasseHangenBrustlin, Inc.
- Wardrop, J., (1952) Some Theoretical Aspects of Road Traffic Research, *Proceedings of the Institution of Civil Engineers*, Part 2, 325–378.
- Weintraub, A., Ortiz, C., and Gonzales, J., (1985) Accelerating Convergence of the Frank-Wolfe Algorithm. *Transportation Research Part B*, 19(2), 113–122.
- Wolfe, P., (1970) Convergence Theory in Nonlinear Programming. In *Integer and Nonlinear Programming* (J. Abadie, ed.), North-Holland, Amsterdam, Netherlands, 1–36.